



TUGAS AKHIR - KS 141501

**RANCANG BANGUN RESTFUL WEB SERVICE PADA
SISTEM REKOMENDASI E-COMMERCE BERBASIS GRAF
NEO4J DENGAN METODE COLLABORATIVE FILTERING
(STUDI KASUS: FORBENTO)**

***RESTFUL WEB SERVICE DEVELOPMENT ON
ECOMMERCE RECOMMENDER SYSTEM BASED ON
GRAPH DATABASE NEO4J USING COLLABORATIVE
FILTERING (STUDY CASE: FORBENTO)***

MARINA SAFITRI

NRP 5213 100 090

Dosen Pembimbing :

Radityo Prasetyanto W., S.Kom, M.Kom

JURUSAN SISTEM INFORMASI

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

RANCANG BANGUN RESTFUL WEB SERVICE PADA SISTEM REKOMENDASI E-COMMERCE BERBASIS GRAF NEO4J DENGAN METODE COLLABORATIVE FILTERING (STUDI KASUS: FORBENTO)

MARINA SAFITRI
NRP 5213 100 090

Dosen Pembimbing:
Radityo Prasetyanto W., S.Kom, M.Kom

Halaman ini sengaja dikosongkan



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS 141501

RESTFUL WEB SERVICE DEVELOPMENT ON ECOMMERCE RECOMMENDER SYSTEM BASED ON GRAPH DATABASE NEO4J USING COLLABORATIVE FILTERING (STUDY CASE: FORBENTO)

MARINA SAFITRI

NRP 5213 100 090

Dosen Pembimbing:

Radityo Prasetyanto W., S.Kom, M.Kom

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

RANCANG BANGUN RESTFUL WEB SERVICE PADA SISTEM REKOMENDASI E-COMMERCE BERBASIS GRAF NEO4J DENGAN METODE COLLABORATIVE FILTERING (STUDI KASUS: FORBENTO)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

MARINA SAFITRI
NRP. 5213 100 090

Surabaya, Juli 2017

DEPARTEMEN SISTEM INFORMASI



Dr. Ir. Aris Prahyanto, M.Kom.
NIP.196503101991021001

LEMBAR PERSETUJUAN

RANCANG BANGUN RESTFUL WEB SERVICE PADA SISTEM REKOMENDASI *E-COMMERCE* BERBASIS GRAF NEO4J DENGAN METODE COLLABORATIVE FILTERING (STUDI KASUS: FORBENTO)

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Oleh :

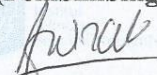
MARINA SAFITRI
NRP. 5213 100 090

Disetujui Tim Penguji : Tanggal Ujian : 11 Juli 2017
Periode Wisuda : September 2017


Radityo Prasetyanto W., S.Kom, M.Kom.


(Pembimbing I)

Faizal Johan Atletiko, S.Kom., M.T.


(Penguji I)

Renny Pradina Kusumawardani S.T., M.T.


(Penguji II)

RANCANG BANGUN RESTFUL WEB SERVICE PADA SISTEM REKOMENDASI *E-COMMERCE* BERBASIS GRAF NEO4J DENGAN METODE COLLABORATIVE FILTERING (STUDI KASUS: FORBENTO)

Nama Mahasiswa : Marina Safitri
NRP : 5213 100 090
Jurusan : Sistem Informasi
Pembimbing 1 : Radityo Prasetyanto W.,
S.Kom, M.Kom

ABSTRAK

Tingginya potensi dan peluang yang menjanjikan dari bisnis online di Indonesia berdampak pada pemanfaatan open source e-commerce berbasis CMS yang semakin tinggi dengan keberagaman platform yang digunakan. Implementasi sistem rekomendasi menjadi hal yang penting dalam sebuah e-commerce untuk meningkatkan pelayanan bisnis. Namun tidak semua platform atau website penjualan pribadi yang digunakan menyediakan sistem rekomendasi. Diperlukan berbagai penyesuaian teknologi untuk menerapkan sistem rekomendasi pada sistem yang berbeda, selain itu sistem rekomendasi yang ada saat ini masih menggunakan representasi data dari relational database yang kurang efektif dalam menerjemahkan relasi antar data dan pemrosesan rekomendasi.

Penelitian sebelumnya mampu menghasilkan rekomendasi yang lebih efektif & membuktikan pemrosesan yang lebih baik dengan menggunakan basis data graf serta memberikan hasil rekomendasi yang akurat dengan menggunakan metode collaborative filtering dalam membuat rekomendasi.

Dalam penelitian ini, dilakukan pembangunan RESTful Web Service rekomendasi produk untuk e-commerce yang dibangun dengan teknologi Node.js dan basis data graf neo4j dengan menerapkan metode user-based collaborative filtering

dalam menghasilkan rekomendasi. Dengan pengujian pada lingkungan klien woocommerce dan menggunakan data transaksi forbento, didapatkan ketepatan rekomendasi sebesar 85,8% dari 4 buah skenario dan 16 test case.

Kata Kunci:

Sistem Rekomendasi, Basis Data Graf, Neo4j, Collaborative Filtering

RESTFUL WEB SERVICE DEVELOPMENT ON ECOMMERCE RECOMMENDER SYSTEM BASED ON GRAPH DATABASE NEO4J USING COLLABORATIVE FILTERING (STUDY CASE: FORBENTO)

Student Name : Marina Safitri
NRP : 5213 100 090
Department : Information System
Supervisor 1 : Radityo Prasetyanto W.,
S.Kom, M.Kom

ABSTRACT

The highly potential and promising opportunities of online business in Indonesia impact on the utilization of CMS-based e-commerce that increasingly high with the diversity of platforms. Implementation of recommendation system becomes important in an e-commerce to improve business services. However, not all personal sales platforms or websites provide recommendation systems. Various technological adjustments are required to implement a recommendation system on each of the different systems, and the current recommendation system still uses a less effective data representation of relational databases in translating relationships between data and recommendation process.

There have been several studies to produce more effective and efficient recommendations that prove better processing by using graph databases. Other studies also provide accurate recommendation results by using collaborative filtering methods in making recommendations.

In this research, the RESTful Web Service product recommendation for e-commerce is developed which is built with Node.js technology and neo4j graph database using

collaborative filtering method in generating recommendations. By testing in the woocommerce client environment and using forbento transaction data, the recommendation accuracy was 85.8% out of 4 scenarios and 16 test cases.

Keyword:

Recommendation System, Graph Database, Neo4j, Collaborative Filtering

KATA PENGANTAR

Puji dan syukur penulis tuturkan ke hadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan dan hidayah-Nya kepada penulis sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir dengan judul:

RANCANG BANGUN RESTFUL WEB SERVICE PADA SISTEM REKOMENDASI *E-COMMERCE* BERBASIS GRAF NEO4J DENGAN METODE COLLABORATIVE FILTERING (STUDI KASUS: FORBENTO)

yang merupakan salah satu syarat kelulusan pada Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materiil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang sudah melauangkan waktu, tenaga dan pikirannya. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada:

1. Bapak Drs.Ec. Moh. Chomsin Hidayat dan Ibu Nur Azizah selaku kedua orang tua serta Rafika Cahyani, Fikri Firmansyah dan Fahmi Hidayat Aziz selaku saudara kandung dari penulis yang tiada henti memberikan dukungan dan semangat.
2. Bapak Radityo Prasetyanto W., S.Kom, selaku dosen pembimbing yang senantiasa meluangkan waktu, memberikan ilmu dan petunjuk, serta memotivasi untuk kelancaran tugas akhir penulis.
3. Ibu Feby Artwodini, S.Kom, MT selaku dosen wali penulis selama menempuh pendidikan di Jurusan Sistem Informasi yang memberi pengalaman &nasehat kepada penulis
4. Bapak Faisal johan Atletiko, S.Kom, M.Kom, Ibu Nur Aini R., S.Kom., M.Sc.Eng., Ph.D., dan Ibu Renny Pradina K.,

- S.T., M.T., SCJP, selaku dosen penguji yang telah memberikan saran dan kritik untuk perbaikan tugas akhir.
5. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang bermanfaat kepada penulis.
 6. Stezar Priansya dan Harun Rizal, selaku sahabat yang telah memberikan pencerahan dan membimbing penulis saat mengalami kesulitan
 7. Risa Perdana, Stezar Priansya, Shania Olivia, Pramita Lucianna, Provani Winda, Delina Rahayu, Chandra Surya, Ikhwan Aziz, M. Fahmi, Nadya Chandra, Caesar Gilang, Rani Oktavia, Bintang Setyawan, Tetha Valianta, Alvin Rahman yang telah mendukung dan menemani penulis dari masa mahasiswa baru hingga tugas akhir ini dapat diselesaikan.
 8. Tetha, Alvin, Wisnu, Risa, Stezar, Niko, Adnan, Harun, Lutfi, Kusnanta, serta para penghuni laboratorium ADDI yang telah menemani pengerjaan tugas akhir ini selama di laboratorium.
 9. Rekan-rekan HMSI Muda Berkarya dan BELTRANIS yang telah memberikan banyak pembelajaran, kenangan manis dan pahit semasa kuliah.
 10. Berbagai pihak yang tidak bisa disebutkan satu persatu yang telah turut serta menyukseskan penulis dalam menyelesaikan tugas akhir.

Penyusunan laporan ini masih jauh dari kata sempurna sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 03 Juli 2017

Penulis,

Marina Safitri

DAFTAR ISI

ABSTRAK.....	i
ABSTRACT.....	iii
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xv
DAFTAR KODE.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
1.6 Relevansi Penelitian	5
BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian Sebelumnya	7
2.2 Dasar Teori.....	9
2.2.1 E-commerce.....	9
2.2.2 Sistem Arsitektur SaaS	9
2.2.3 RESTFull Web Service.....	10
2.2.4 Sistem Rekomendasi.....	12
2.2.5 Collaborative Filtering.....	13
2.2.6 Cosine Similarity	16
2.2.7 Euclidian Distance	17
2.2.8 Basis Data Graf.....	17
2.2.9 Node.js	20
2.2.10 Iterative and Incremental Development	22
BAB III METODOLOGI PENELITIAN	25
3.1 Metode Pengerjaan	26
3.1.1 Studi Literatur.....	26
3.1.2 Perancangan Perangkat Lunak.....	26
3.1.3 Penulisan buku tugas Akhir	37
BAB IV PERANCANGAN	39

4.1	Perancangan Data	39
4.1.1	Pengumpulan Data.....	39
4.1.2	Pemilihan Atribut/Analisis kebutuhan data....	44
4.1.3	Pra-pemrosesan Data	45
4.2	Perancangan aplikasi klien	51
4.2.1	Analisis kebutuhan aplikasi klien	51
4.2.2	Desain Sistem	52
4.2.3	Perancangan database klien	53
4.2.4	Perancangan pengujian klien	57
4.3	Perancangan aplikasi server.....	57
4.3.1	Analisis kebutuhan aplikasi server.....	58
4.3.2	Desain sistem server	59
4.3.3	Perancangan database server.....	66
4.3.4	Perancangan library pada server	70
4.3.5	Perancangan pengujian web service.....	71
4.4	Perancangan rekomendasi	71
4.4.1	Perancangan algoritma rekomendasi.....	71
4.4.2	Perancangan pengujian rekomendasi	72
BAB V IMPLEMENTASI		73
5.1	Lingkungan Implementasi	73
5.2	Pembangunan Client.....	74
5.2.1	Instalasi Wordpress.....	75
5.2.2	Instalasi Woocommerce.....	76
5.2.3	Migrasi Data	78
5.2.4	Pembuatan Plugin Import Customer	79
5.2.5	Pembuatan Plugin Import Order	81
5.2.6	Menampilkan rekomendasi.....	82
5.3	Konfigurasi Database	85
5.3.1	Import Data.....	87
5.4	Pembangunan web service.....	88
5.4.1	Pembuatan API.....	89
5.4.2	Pembuatan mekanisme update data	97
5.4.3	Mendapatkan hasil rekomendasi	110
5.4.4	Pembuatan website Recommerce.....	112
BAB VI HASIL DAN PEMBAHASAN		129
6.1	Hasil	129
6.1.1	Pengujian Fungsional (Functional Testing) .	129

6.1.2	Pengujian Rekomendasi.....	137
6.2	Pembahasan.....	173
	Analisis Hasil BlackBox testing	173
	Analisis Hasil Rekomendasi	174
BAB VII KESIMPULAN DAN SARAN.....		181
7.1	Kesimpulan	181
7.2	Saran.....	182
DAFTAR PUSTAKA		185
LAMPIRAN A.....		1
A.1.	Lembar Persetujuan Narasumber	1
A.2.	Pengujian user 1	2
A.3.	Pengujian user 2	3
A.4.	Pengujian user 3	4
A.4.	Pengujian user 3	5

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 1.1. Grafik persebaran penggunaan platform E-commerce di Indonesia.....	2
Gambar 2.1. Arsitektur SaaS.....	10
Gambar 2.2. Arsitektur Restful Web Service	11
Gambar 2.3. Jenis metode pada sistem rekomendasi.....	13
Gambar 2.4. Ilustrasi proses collaborative filtering.....	14
Gambar 2.5. Jenis pendekatan metode Collaborative Filtering	14
Gambar 2.6. Ilustrasi rekomendasi dengan metode user based CF	16
Gambar 2.7. Model basis data graf.....	18
Gambar 2.8. Struktur Cypher Query Language	19
Gambar 3.1. Alur metodologi Penelitian.....	25
Gambar 3.2. Fase Iterative Development	27
Gambar 3.3. Model 2 Layer Graph Database	29
Gambar 3.4. Permodelan data rekomendasi	30
Gambar 3.5. Desain Infrastruktur Web Service Rekomendasi	31
Gambar 3.6. Perancangan desain UI hasil implementasi	34
Gambar 4.1. File csv data produk.....	40
Gambar 4.2. Contoh data produk dalam format csv	40
Gambar 4.3. Tabel produk dalam database.....	41
Gambar 4.4. File csv data pengiriman	41
Gambar 4.5. Contoh data pengiriman dalam format csv	42
Gambar 4.6. Tabel pengiriman dalam database.....	42
Gambar 4.7. File csv data transaksi	43
Gambar 4.8. Contoh data transaksi dalam format csv	43
Gambar 4.9. Tabel pengiriman dalam database.....	44
Gambar 4.10. Hasil penyaringan data	46
Gambar 4.11. Data Pelanggan yang belum dibersihkan	47
Gambar 4.12. Contoh data yang tidak valid	48
Gambar 4.13. Hasil penggabungan data	50
Gambar 4.14. Database gabungan pelanggan dan order	50

r 4.15. Data pembelian beserta recurrence order	51
Gambar 4.16. Desain use case client	53
Gambar 4.17. Swimlane generate rekomendasi pada server ..	59
Gambar 4.18. Swimlane Update data server	60
Gambar 4.19. Desain use case server	61
Gambar 4.20. Halaman Home section top_content	62
Gambar 4.21. Halaman home section feature	62
Gambar 4.22. Halaman home section how to work	62
Gambar 4.23. Halaman Register Member	63
Gambar 4.24. Halaman Login Member	63
Gambar 4.25. Halaman Login Toko	64
Gambar 4.26. Halaman Dashboard Awal	64
Gambar 4.27. Halaman Dashboard section product-information	65
Gambar 4.28. Halaman Dashboard section update data.....	65
Gambar 4.29. Halaman Profil Toko Member	65
Gambar 4.30. Halaman Daftar Produk	66
Gambar 4.31. Desain Model Graph	70
Gambar 5.1. Direktori Client	75
Gambar 5.2. Halaman Konfigurasi awal wordpress	76
Gambar 5.3. Install plugin woocommerce	77
Gambar 5.4. tampilan sukses install woocommerce	77
Gambar 5.5. Tampilan menu bar install woocommerce	78
Gambar 5.6. File sql wodpress forbento	78
Gambar 5.7. Import file sql ke database wordpress	79
Gambar 5.8. Install plugin import user from CSV with meta	79
Gambar 5.9. Contoh data CSv yang telah diolah	80
Gambar 5.10. Tampilan plugin import CSV	80
Gambar 5.11. Mengaktifkan database neo4j.....	85
Gambar 5.12. Install driver neo4j pada node js	86
Gambar 5.13. Konfigurasi koneksi bolt pada neo4j.....	86
Gambar 5.14. Hasil Import data produk	87
Gambar 5.15. Hasil import data pelanggan	88
Gambar 5.16. Hasil import data order	88
Gambar 5.17. Direktori aplikasi server	89
Gambar 5.18. Contoh file yang dibuat sistem proses update .	94
Gambar 5.19. Contoh data produk dala format json	95

Gambar 5.20. Contoh data pelanggan dalam format jsc	
Gambar 5.21. Contoh data order dalam format json.....	135
Gambar 6.1. Rekomendasi Produk saat tidak ada user login	135
Gambar 6.2. Console server saat mengambil rekomendasi	135
Gambar 6.3. Rekomendasi produk IbuFera(id cust:880)	136
Gambar 6.4. Perbandingan ketepatan rekomendasi	174
Gambar 6.5. Pengaruh Metode perhitungan kesamaan user dengan hasil rekomendasi	176
Gambar 6.6. pengaruh hasil rekomendasi terhadap tipe pelanggan	177
Gambar 6.7. Pengaruh recurrence order pada hasil rekomendasi	178
Gambar A.1. Lembar Persetujuan Narasumber	1
Gambar A.2. Lembar Pengujian user 1	2
Gambar A.3. lembar Pengujian User 3	3
Gambar A.4. Lembar Pengujian User 4.....	4

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1. Penelitian terdahulu	7
Tabel 2.2. Detail metode pada REST	11
Tabel 3.1. Daftar node pada graf	30
Tabel 3.2. Daftar Relationship pada graf	31
Tabel 4.1. Data dictionary wordpress	54
Tabel 4.2. Data dictionary woocommerce	56
Tabel 4.3. Daftar Node dalam server.....	66
Tabel 4.4. Node member dalam graph.....	67
Tabel 4.5. Node Shop dalam graph	67
Tabel 4.6. Node customer dalam graph	68
Tabel 4.7. Node produk dalam graph	68
Tabel 4.8. Hubungan antar graph	68
Tabel 4.9. Daftar library node js yang akan digunakan	70
Tabel 5.1. Spesifikasi lingkungan implementasi	73
Tabel 5.2. Lingkungan teknologi yang digunakan.....	73
Tabel 5.3. query cypher - proses mengecek toko.....	98
Tabel 5.4. query cypher - menentukan offset	99
Tabel 5.5. cypher query - menentukan offset order	101
Tabel 5.6. Penjelasan cypher proses update pelanggan baru	103
Tabel 5.7. perhitungan cosine similarity query cypher	107
Tabel 5.8. Penjelasan perhitungan euclidian distance dalam query cypher.....	109
Tabel 5.9. Penjelasan query cypher pada proses mengambil rekomendasi	111
Tabel 5.10. Input parameter pada halaman register	114
Tabel 5.11. Cypher query saat registrasi	115
Tabel 5.12. Parameter input form login member	116
Tabel 5.13. Cypher query login member	118
Tabel 6.1. fitur Recommerce	129
Tabel 6.2. Pengujian Fungsional Aplikasi	130
Tabel 6.3. Data dan riwayat pembelian pelanggan id 880 ...	154
Tabel 6.4. Data dan riwayat pembelian pelanggan id 52	155
Tabel 6.5. Data dan riwayat pembelian pelanggan id 18	155
Tabel 6.6. Data dan riwayat pembelian pelanggan id 952 ...	155
Tabel 6.7. Pengujian set rekomendasi test case 1	158

Tabel 6.8. Ringkasan hasil pengujian test case 1	159
Tabel 6.9. Hasil uji set rekomendasi test case 2.....	159
Tabel 6.10. Ringkasan hasil pengujian test case 2	160
Tabel 6.11. Hasil uji set rekomendasi test case 3.....	160
Tabel 6.12. Ringkasan hasil pengujian test case 3	161
Tabel 6.13. Hasil uji set rekomendasi test case 4.....	161
Tabel 6.14. Ringkasan hasil pengujian test case 4	162
Tabel 6.15. Hasil uji set rekomendasi test case 5.....	162
Tabel 6.16. Ringkasan hasil pengujian test case 5	163
Tabel 6.17. Hasil uji set rekomendasi test case 6.....	163
Tabel 6.18. Ringkasan hasil pengujian test case 6	164
Tabel 6.19. Hasil uji set rekomendasi test case 7.....	164
Tabel 6.20. Ringkasan hasil pengujian test case 7	164
Tabel 6.21. Hasil uji set rekomendasi test case 8.....	165
Tabel 6.22. Ringkasan hasil pengujian test case 8	165
Tabel 6.23. Hasil uji set rekomendasi test case 9.....	166
Tabel 6.24. Ringkasan hasil pengujian test case 9	166
Tabel 6.25. Hasil uji set rekomendasi test case 10.....	167
Tabel 6.26. Ringkasan hasil pengujian test case 10	167
Tabel 6.27. Hasil uji set rekomendasi test case 11.....	168
Tabel 6.28. Ringkasan hasil pengujian test case 11	168
Tabel 6.29. Hasil uji set rekomendasi test case 12.....	169
Tabel 6.30. Ringkasan hasil pengujian test case 12	169
Tabel 6.31. Hasil uji set rekomendasi test case 13.....	170
Tabel 6.32. Ringkasan hasil pengujian test case 13	170
Tabel 6.33. Hasil uji set rekomendasi test case 14.....	171
Tabel 6.34. Ringkasan hasil pengujian test case 14	171
Tabel 6.35. Hasil uji set rekomendasi test case 15.....	172
Tabel 6.36. Ringkasan hasil pengujian test case 15	172
Tabel 6.37. Hasil uji set rekomendasi test case 16.....	173
Tabel 6.38. Ringkasan hasil pengujian test case 16	173
Tabel 6.39. Persentase hasil uji dengan cosine	175
Tabel 6.40. Persentase hasil uji dengan cosine & recurrence order	175
Tabel 6.41. Persentase hasil uji dengan euclidian.....	175
Tabel 6.42. Persentase hasil uji dengan euclidian & recurrence order	176

DAFTAR KODE

Kode 4.1. Query untuk filter informasi pelanggan	46
Kode 4.2. Query untuk menghilangkan simbol dan karakter	48
Kode 4.3. Query untuk menghilangkan duplikasi data	49
Kode 4.4. Query penggabungan data pelanggan dan order.....	49
Kode 4.5. mencari recurrence order	51
Kode 5.1. Query untuk mengosongkan tabel.....	79
Kode 5.2. Konfigurasi awal plugin.....	81
Kode 5.3. Menampilkan rekomendasi pada klien.....	83
Kode 5.4. Membaca hasil rekomendasi dari server	84
Kode 5.5. Menampilkan rekomendasi sebagai featured product	85
Kode 5.6. Membuat instance library driver neo4j	86
Kode 5.7. Menyimpan data untuk pembuatan API.....	90
Kode 5.8. Pemilihan data produk untuk woocommerce API	91
Kode 5.9. Pemuatan data pelanggan dalam woocommerce API	92
Kode 5.10. Memuat data order pada woocommerce API	92
Kode 5.11. Pembuatan File token dalam format json	92
Kode 5.12. Callback function dataCustomer	93
Kode 5.13. Mencari node toko berdasarkan token.....	98
Kode 5.14. Menghitung nilai offset pelanggan.....	99
Kode 5.15. Menghitung nilai offset produk.....	100
Kode 5.16. Menghitung nilai offset order	101
Kode 5.17. Mengupdate pelanggan baru pada graph.....	103
Kode 5.18. Update produk baru pada graph	104
Kode 5.19. Update order baru pada graph	105
Kode 5.20. Menghitung recurrence order sebagai nilai relationship.....	105
Kode 5.21. Menambahkan nilai kesamaan pelanggan dengan perhitungan cosine similarity	106
Kode 5.22. mencari rekomendasi yang tepat berdasarkan collaborative filtering	111
Kode 5.23. Implementasi halaman home recommerce	113

Kode 5.24. Implementasi halaman registrasi recommerce..	
Kode 5.25. Melakukan rendering halaman register	114
Kode 5.26. Insert node member setelah registrasi.....	115
Kode 5.27. Implementasi halaman login member	116
Kode 5.28. Rendering halaman login member	117
Kode 5.29. Proses login dengan graph	118
Kode 5.30. Implementasi halaman pendaftaran toko	120
Kode 5.31. Rendering halaman pendaftaran toko	120
Kode 5.32. proses insert toko baru pada graph	121
Kode 5.33. Implementasi halaman login toko	122
Kode 5.34. Rendering halaman login toko	122
Kode 5.35. proses login toko pada graph.....	123
Kode 5.36. Implementasi halaman dahboard utama	124
Kode 5.37. Rendering halaman dashboard	124
Kode 5.38. Mengambil data toko berdasarkan token.....	125
Kode 5.39. parameter yang didapatkan dari halaman dahboard	125
Kode 5.40. Implementasi halaman profil member	126
Kode 5.41. Paramater yang disimpan pada halaman user	126
Kode 5.42. Parameter yang di simpan pada halaman tabel data	127
Kode 6.1. Query cypher rekomendasi skenario 1	156
Kode 6.2. Query cypher rekomendasi skenario 2	157
Kode 6.3. Query cypher rekomendasi skenario 3	157
Kode 6.4. Query cypher rekomendasi skenario 4	158

BAB I

PENDAHULUAN

Pada bab pendahuluan akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami

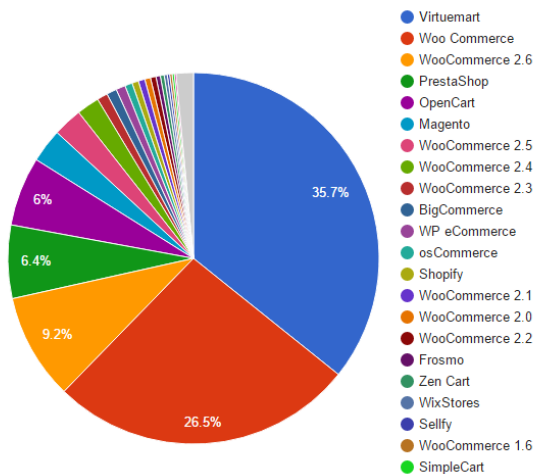
1.1 Latar Belakang

Penetrasi internet yang kini telah digunakan oleh lebih dari 132,7 juta pengguna di Indonesia[1] telah memberikan banyak manfaat di berbagai bidang, salah satu diantaranya adalah pada sektor bisnis online atau *e-commerce*. Hal ini dibuktikan dengan potensi bisnis online dengan nilai penjualan global *e-commerce* tahun 2015 yang hampir mencapai 1 triliun dolar atau tumbuh sebesar 18% dibandingkan tahun 2014 menurut riset dan penelitian firma konsultan bisnis dan manajemen AT Kearny [2].

Tingginya potensi dan peluang yang menjanjikan dari bisnis online yang di Indonesia, pemanfaatan *open source e-commerce* berbasis CMS (*Content Management System*) yang tersebar luas di internet menjadi andalan para pelaku bisnis dalam meningkatkan penjualan dan pelayanannya kepada pelanggan. Adanya web *e-commerce* diharapkan dapat memudahkan pelanggan saat berbelanja. Tidak hanya itu untuk membuat pelanggan tetap loyal, diperlukan layanan *e-commerce* yang mampu memperlakukan setiap pelanggannya secara spesial, salah satunya adalah sistem rekomendasi. Sistem rekomendasi memberikan usulan produk yang terkustomisasi sesuai ketertarikan setiap pelanggan secara unik. Namun, tidak semua platform atau website penjualan pribadi yang digunakan

menyediakan sistem rekomendasi sehingga dapat membatasi layanan *e-commerce* yang digunakan.

Setiap bisnis menggunakan platform yang berbeda sesuai dengan kebutuhan dan memiliki alasannya masing-masing dalam pemilihan platform yang sesuai dengan bisnis yang dijalankan. Beragamnya penggunaan platform *e-commerce* di Indonesia dapat ditunjukkan pada bagan 1. Dari Alexa 1 Million Top Site Survey, diketahui bahwa hingga tahun 2017 Virtuemart menjadi platform yang paling populer dan banyak digunakan di Indonesia hingga mencapai 35,70% dan mampu menguasai pangsa pasar secara stabil disusul dengan woocommerce, prestashop, opencart hingga magento di peringkat selanjutnya [3].



Gambar 1.1. Grafik persebaran penggunaan platform E-commerce di Indonesia

Sumber: Alexa 1 Million Top Site Survey, 2017

Tentu akan sangat tidak efisien jika developer harus membuat sistem rekomendasi satu per satu pada setiap platform yang berbeda. Terlebih lagi, penelitian sebelumnya menemukan keterbatasan pada implementasi rekomendasi berbasis relational database yang kurang efektif dalam

merepresentasikan informasi & hubungan antara produk dan pelanggan dengan segala transaksi yang terjadi [4][5].

Oleh karena itu, tugas akhir ini bertujuan untuk mengatasi berbagai permasalahan tersebut dengan mengajukan konsep pembangunan *restful web service* rekomendasi sebagai layanan SaaS (Software as a Service) dengan menggunakan basis data graf *neo4j* untuk menghasilkan representasi informasi yang lebih baik. Hal ini dilakukan karena permodelan graf pada basis data relational akan menyebabkan masalah terkait waktu pemrosesan yang besar seiring jumlah data yang besar dalam situs *e-commerce*[5]. Basis data graf *neo4j* adalah sebuah basis data yang mengutamakan hubungan antar entitas, penggunaan basis data ini telah dioptimasi sedemikian rupa sehingga memiliki performansi yang lebih baik[6]. Tugas akhir ini menggunakan metode *collaborative filtering* yang digunakan sebagai algoritma dalam pembangunan sistem rekomendasi untuk menghasilkan hasil personalisasi dan usulan produk yang sesuai dengan karakteristik setiap pelanggan yang unik berdasarkan korelasi antar pelanggan (*customer based*).

Layanan sistem rekomendasi dalam tugas akhir ini dibangun dengan menyediakan REST API yang dibangun dengan teknologi Node.js dan *neo4j* yang dipilih karena fleksibilitas dan kemudahannya dalam berkomunikasi satu sama lain sehingga memungkinkan berbagai pihak dapat dengan mudah menggunakannya melalui berbagai platform. Dengan adanya layanan rekomendasi ini diharapkan dapat memberikan kemudahan bagi setiap pengguna *e-commerce* untuk mengetahui rekomendasi dari berbagai data transaksi yang berbeda dengan pemrosesan yang lebih efisien melalui basis data graf.

1.2 Perumusan Masalah

Berdasarkan uraian latar belakang maka didapatkan fokus rumusan masalah dalam penelitian berikut yaitu:

1. Bagaimana merancang dan membangun restful web service sistem rekomendasi?
2. Bagaimana melakukan permodelan data menggunakan basis data graf neo4j dalam melakukan simulasi transaksi penjualan dengan algoritma collaborative filtering?
3. Bagaimana kesesuaian rekomendasi yang dihasilkan dengan studi kasus toko online forbento?

1.3 Batasan Masalah

Adapun penelitian berikut ini terbatas pada:

1. Implementasi algoritma collaborative filtering yang digunakan adalah customer based collaborative filtering
2. Permodelan basis data graf menggunakan Neo4j dengan implementasi CQL (Cypher Query Language)
3. Data yang digunakan untuk pengujian ialah data transaksi penjualan pada online shop forbento.
4. Pengujian penelitian ini terbatas pada pengujian fungsionalitas sistem dan ketepatan rekomendasi
5. Aplikasi client yang digunakan untuk pengujian adalah platform woocommerce
6. Perhitungan similarity user yang digunakan terbatas pada metode cosine similarity dan euclidian distance.

1.4 Tujuan Penelitian

Berdasarkan perumusan masalah dan batasan masalah tujuan dari penelitian tugas akhir berikut ini ialah:

1. Membuat sistem rekomendasi sebagai SaaS melalui pemrosesan basis data graf neo4j dengan yang dapat diimplementasikan ke dalam berbagai *e-commerce*.
2. Mengetahui proses permodelan data yang tepat untuk sistem rekomendasi berbasis graf
3. Menganalisis penerapan algoritma collaborative filtering dalam menghasilkan rekomendasi yang tepat.

1.5 Manfaat Penelitian

Tugas Akhir berikut ini diharapkan dapat memberikan beberapa manfaat sebagai berikut:

Bagi Penulis

Mendapatkan pengetahuan mengenai cara merancang sistem rekomendasi rest web service, melakukan permodelan data ke dalam sistem graf serta pengetahuan terkait cara kerja algoritma dalam menghasilkan insight yang bermanfaat bagi sistem penjualan *e-commerce*.

Bagi Developer E-commerce:

Memberikan kemudahan dalam mengimplementasikan sistem rekomendasi pada platform *e-commerce* dengan berbagai teknologi, sistem informasi atau bahasa pemrograman tanpa melakukan banyak kustomisasi kode dalam penerapan algoritma rekomendasi.

Bagi Pelaku Bisnis:

Meningkatkan kualitas *e-commerce* dalam memberikan pelayanan kepada pelanggan, membantu meningkatkan loyalitas pelanggan melalui sistem rekomendasi saat berbelanja melalui channel *e-commerce*

1.6 Relevansi Penelitian

Tugas akhir ini berkaitan dengan Laboratorium ADDI dengan mata kuliah yang berkaitan yaitu Desain Basis Data, Manajemen & Administrasi Basis Data, Statistika, Sistem Cerdas, Konstruksi Perangkat Lunak, Pemrograman berbasis Web, Penggalan Data dan Analitika Bisnis, Sistem Pendukung Keputusan, dan Kecerdasan Bisnis.

Halaman ini sengaja dikosongkan

BAB II

TINJAUAN PUSTAKA

Pada bab tinjauan pustaka akan menjelaskan beberapa penelitian terdahulu yang terkait dengan tugas akhir berikut ini dan beberapa teori terkait dengan tugas akhir berikut.

2.1 Penelitian Sebelumnya

Bagian ini akan menjelaskan penelitian sebelumnya yang terkait dengan Tugas Akhir berikut ini dan hasil yang diperoleh penulis dari penelitian tersebut yang dijelaskan pada table 1 penelitian sebelumnya berikut ini.

Tabel 2.1. Penelitian terdahulu

<p>Paper No: 1 Judul: A Graph Model for <i>E-commerce</i> Recommender System Pengarang/tahun/sumber: Huang, Zan Chung; Wingyan Chen, Hsinchun Metode: Permodelan data dengan uji fungsionalitas & fleksibilitas menggunakan direct retrieval, association mining, dan high-degree association retrieval. Kesimpulan:</p> <ul style="list-style-type: none">• Penelitian ini mengajukan pembangunan model graf yang umum untuk rekomendasi produk pada <i>e-commerce</i>.• Dalam hal ini customer, product dan transaksi dimodelkan pada pengembangan graf yang terdiri dari 2 layer. Untuk menguji efektifitas dan fleksibilitas model, dilakukan 3 percobaan metode untuk membuat rekomendasi dengan 3 pendekatan yaitu content-based, collaborative dan hybrid• Penelitian ini telah membuktikan bahwa model yang dibuat telah sesuai untuk merepresentasikan data berbagai kombinasi dalam pendekatan <i>e-commerce</i> sehingga dapat menjadi rujukan dalam penelitian ini
<p>Paper No: 2 Judul: A Graph Model for Recommender Systems Pengarang/tahun/sumber: Chen, Hong Gan; Mingxin Metode: Pengembangan algoritma permodelan graf</p>

Kesimpulan:

Berdasarkan hasil penelitian, dengan mengembangkan algoritma permodelan graf, yang menggunakan perhitungan kemiripan item dan pemilihan hubungan antara pengguna dan item produk memberikan hasil:

- Dari tiga metode perhitungan similarity, cosine dan pearson memberikan hasil yang lebih baik daripada jaccard
- Dibandingkan matriks kedekatan dengan range 0-1, data input rating dengan range 1-5 lebih efektif untuk algoritma rekomendasi
- Model graf bipartite tidak hanya mampu merepresentasikan fleksibilitas data, namun juga mengurangi dampak yang disebabkan karena faktor kejarangan data.

Paper No: 3

Judul: Sistem Rekomendasi Jual Beli Barang Dengan Memanfaatkan Metode Collaborative Filtering dan Basis Data Graf . Studi Kasus : Bukalapak . com

Pengarang/tahun/sumber: Wibowo, David Eko (2013) [5]

Metode: Proses perhitungan kemiripan antar item dilakukan dengan metode KNN, sedangkan untuk menghasilkan rekomendasi menggunakan algoritma grafil dengan pendekatan metode *content-based filtering*

Kesimpulan:

Deri penelitian yang dilakukan dan diterapkan pada sistem bukalapa menghasilkan kesimpulan berikut:

- Dari data training dan data test dalam pengujian, dihasilkan presisi maksimal 33,64% dan recall maksimal 51,03% untuk algoritma grafil yang cocok untuk diimplementasikan di bukalapak.
- Permodelan graf berdasarkan interaksi pengguna berpengaruh pada tipe algoritma yang digunakan
- Waktu yang dibutuhkan algoritma grafil dalam menghasilkan rekomendasi tanpa memperhitungkan preproses adalah 324 ms, dan 469 ms untuk algoritma KNN.

Paper No: 4

Judul: Enhancing the Accuracy of Recommender System Using Graph Databases

Pengarang/tahun/sumber: Sharma, Ashish (2015)[7]

Metode: Kombinasi metode dari database graf dan LSH (Locality Sensitive Hashing)

Kesimpulan:

Dari penelitian ini, dihasilkan model baru untuk penyimpanan database graf bagi implementasi rekomendasi berbasis collaborative filtering yang mampu menyelesaikan permasalahan kejarangan data dengan menggunakan struktur data Locality Sensitive Hashing. Dari eksperimen juga dihasilkan hasil eksekusi rekomendasi berbasis graf yang lebih cepat dan efisien dibandingkan dengan relational database dengan perbedaan hingga 5000 ms

2.2 Dasar Teori

Pada bagian ini memaparkan mengenai teori-teori yang berkaitan dengan Tugas Akhir yang dilakukan oleh penulis.

2.2.1 E-commerce

Electronic Commerce (Perniagaan Elektronik), sebagai bagian dari Electronic Business (bisnis yang dilakukan dengan menggunakan electronic transmission), oleh para ahli dan pelaku bisnis dicoba dirumuskan definisinya. Secara umum e commerce dapat didefinisikan sebagai segala bentuk transaksi perdagangan/perniagaan barang atau jasa (trade of goods and service) dengan menggunakan media elektronik. *e-commerce* adalah penggunaan jaringan computer untuk melakukan komunikasi bisnis dan transaksi komersial. [8]

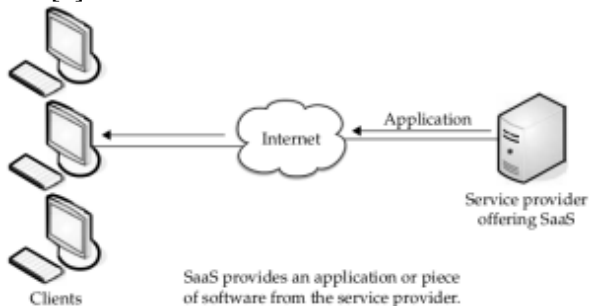
2.2.2 Sistem Arsitektur SaaS

Software as a Service (SaaS) memiliki peran penting baik model bisnis dan model pengiriman aplikasi. SaaS memungkinkan pelanggan untuk dapat memanfaatkan aplikasi berbasis pay-as-you-go dan menghilangkan kebutuhan untuk melakukan instalasi serta menjalankan aplikasi pada perangkat

keras sendiri. Pelanggan umumnya mengakses aplikasi melalui browser Web atau *thin client* melalui Internet. SaaS umumnya berbasis langganan dan semua dukungan yang berkelanjutan, pemeliharaan, dan upgrade disediakan oleh perangkat lunak milik vendor sebagai bagian dari layanan tersebut. Kemampuan kustomisasi aplikasi, jika tersedia, umumnya diberikan kepada semua pelanggan secara konsisten.

2.2.3 RESTFull Web Service

REST (REpresentational State Transfer) merupakan standar arsitektur komunikasi berbasis web yang diterapkan dalam pengembangan layanan berbasis web. REST sendiri umumnya menggunakan HTTP (Hypertext Transfer Protocol) sebagai protokol untuk komunikasi data. RESTFull Web Service membangun integrasi dengan cara yang lebih ringan dan sederhana [9].

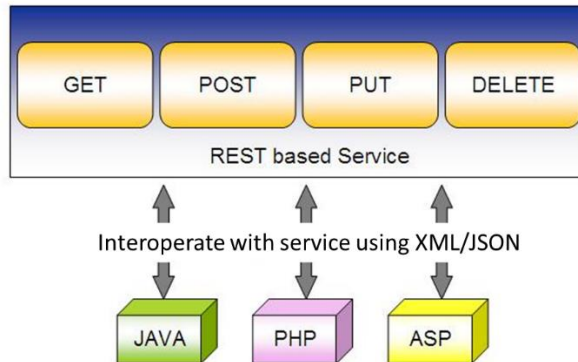


Gambar 2.1. Arsitektur SaaS

Sumber: (Putra, Arfiansyah Dwi, 2012)[10]

Terdapat dua bagian pesan yang digunakan untuk membangun komunikasi dengan server yaitu pesan *Header* dan pesan *Body*. HTTP *Header* meliputi *header request*, *header response* dan terdapat bidang entitas *header*. Setiap *request resource* dari masing-masing client dapat dikendalikan dengan memanfaatkan HTTP Header. Sedangkan HTTP Body meliputi pesan HTTP yang digunakan untuk membuat entitas body melalui protokol HTTP yang berhubungan. Saat client melakukan request, HTTP Body berisi informasi setiap

parameter. REST dapat mengoperasikan operasi CRUD (*create*, *read*, *update*, dan *delete*) yang dapat dilakukan dengan memanfaatkan metode HTTP antara lain POST, GET, PUT, dan DELETE. Berikut ini dijabarkan pemetaan metode HTTP pada REST



Gambar 2.2. Arsitektur Restful Web Service
 Sumber: (Abhishek, 2010)[11]

Tabel 2.2. Detail metode pada REST

Operasi CRUD	HTTP Method	Format Application/x- www-form encoded	HTTP status
Create	POST	HTTP Encoded Form	Status 201 created
Read	GET	None	Status 200 Ok
Update	PUT	HTTP Encoded Form	Status 200 Ok
Delete	DELETE	None	Status 200 Ok

2.2.4 Sistem Rekomendasi

Sistem rekomendasi merupakan software dan suatu teknik yang mampu memberikan usulan produk yang bernilai dan sesuai dengan penggunanya [12][13]. Usulan ini mampu membantu berbagai proses pembuatan keputusan, seperti produk apa yang ingin dibeli, musik apa yang ingin saya dengarkan, berita apa yang harus saya baca, dan lain sebagainya. Sistem ini mampu memberikan layanan khusus bagi setiap pengunjung website dengan memberikan rujukan eksplorasi yang sesuai untuk masing-masing personalisasi pengunjung di masa depan. Dengan kemampuannya, fitur ini sangat membantu pengunjung yang memiliki sedikit pengalaman untuk memilih dan mengevaluasi berbagai produk/produk yang ada di dalam sebuah website. Sistem rekomendasi juga terbukti mampu mengurangi *overload* pada data [14]. Beberapa tahun terakhir, sistem rekomendasi mampu membuktikan eksistensinya dengan kemampuan untuk menyajikan item/produk yang tepat pada orang yang tepat. Ketertarikan terhadap topik sistem rekomendasi terus meningkat mengingat saat ini sistem rekomendasi memegang peranan penting dalam industri bisnis berbasis online seperti Amazon.com, Youtube, Netflix, IMDB dan lain sebagainya. Begitu juga Netflix yang mendapat penghargaan karena peningkatan performance secara substansial dengan adanya sistem rekomendasi [13]. Sistem rekomendasi yang baik dapat meningkatkan pengalaman pengunjung website melalui interaksi yang diberikan oleh website. Dengan demikian, pengunjung dapat menjadi loyal dengan perlakuan khusus yang diberikan dan memperlakukannya selayaknya pengunjung yang bernilai bagi bisnis yang bersangkutan. Pada umumnya terdapat 3 komponen utama dari sebuah rekomendasi [15] yaitu pelanggan (pelanggan dari sebuah sistem rekomendasi dapat memiliki berbagai karakteristik yang berbeda-beda), produk (produk ini merupakan komponen yang direkomendasikan kepada pelanggan atau informasi yang digunakan untuk merekomendasikan produk yang relevan). Dan algoritma kecocokan antara pelanggan (algoritma untuk

pengambilan informasi yang mampu membantu menemukan produk yang memiliki hubungan terdekat dengan pelanggan). Terdapat berbagai macam teknik pembuatan sistem rekomendasi seperti rekomendasi berbasis content-based, rekomendasi berbasis collaborative, knowledge-based, demographic-based hingga rekomendasi hybrid yang diilustrasikan dengan gambar 2.1. berikut[13]



Gambar 2.3. Jenis metode pada sistem rekomendasi

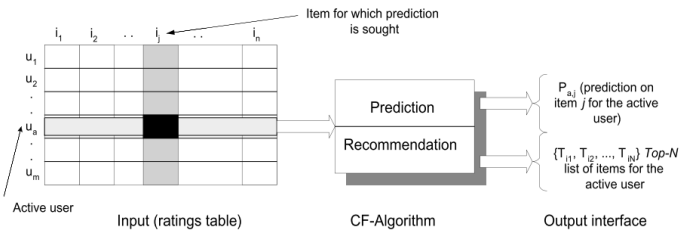
2.2.5 Collaborative Filtering

Collaborative filtering merupakan teknik pemilahan data yang didapatkan dari kolaborasi dari pelanggan lain. Teknik ini merupakan salah satu metode yang paling banyak digunakan untuk membuat rekomendasi bagi pelanggan suatu website yang dipilih karena kemudahannya dan hasil yang rekomendasinya yang baik [16]. Metode ini dipilih berdasarkan fakta bilamana terdapat dua pelanggan, X dan Y yang telah memberikan penilaian terhadap n item/produk atau berperilaku sama dalam lingkungan apapun sehingga akan memberikan penilaian yang sejenis pada barang-barang serupa lainnya juga.

2.2.5.1 Proses Collaborative Filtering

Tujuan utama collaborative filtering adalah menghasilkan produk baru atau sebagai media yang membantu menyajikan produk yang tepat kepada individu tertentu. Pada setiap skenario collaborative filtering terdapat sekumpulan pengguna

U sejumlah m , $U = \{u_1, u_2, \dots, u_m\}$ dan sekumpulan item sejumlah n , $I = \{i_1, i_2, \dots, i_n\}$. Setiap pengguna u memiliki penilaian terhadap item i tertentu. Penilaian tersebut lalu dipetakan ke dalam sebuah matriks rating pengguna terhadap item.



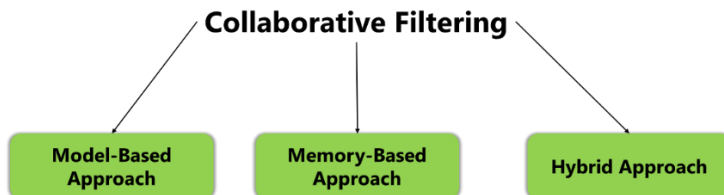
Gambar 2.4. Ilustrasi proses collaborative filtering

Sumber: (Sarwar, Badrul M, 2001)[17]

Pada gambar 2.2 Algoritma collaborative filtering merepresentasikan data pengguna dan item $m \times n$ sebagai matriks rating. Setiap masukan $a_{i,j}$ menggambarkan preferensi pengguna terhadap item. $P_{a,j}$ merupakan prediksi preferensi pengguna target terhadap item yang belum diberikan penilaian olehnya, lalu dihasilkan rekomendasi dari nilai item yang cenderung memiliki nilai prediksi rating tertinggi. [18]

2.2.5.2 Tipe Collaborative Filtering

Collaborative Filtering dibagi menjadi beberapa jenis seperti yang diilustrasikan pada gambar:



Gambar 2.5. Jenis pendekatan metode Collaborative Filtering

Pendekatan model-based pada sistem rekomendasi menggunakan model yang sudah dihasilkan secara offline dan digunakan untuk menghasilkan rekomendasi yang dihasilkan

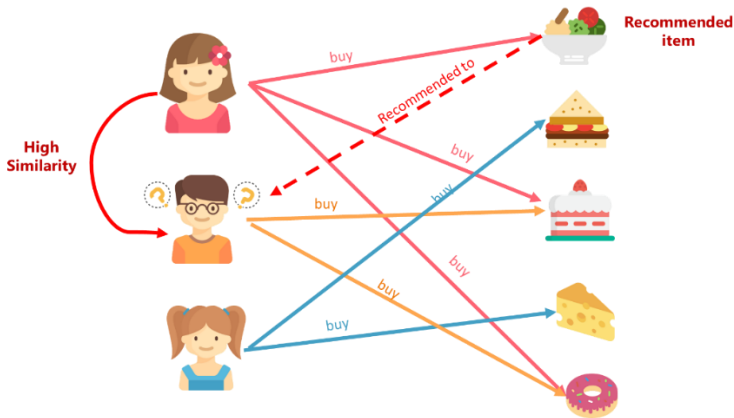
secara online. Pembuatan model dibuat melalui algoritma machine learning yang berbeda-beda seperti Bayesian network, clustering, dan pendekatan rule-based. Dalam hal ini, kekurangan dari rekomendasi berbasis model-based adalah model yang dihasilkan akan memakan waktu yang cukup lama dalam proses perhitungan secara offline. [19]. Di sisi lain, sistem rekomendasi berbasis memory-based menghitung kemiripan antar pengguna atau produk. Disini, seluruh data yang ada di dalam database akan masuk ke dalam memory dan digunakan secara langsung untuk menghasilkan rekomendasi. Sistem ini menggunakan teknik statistika untuk menemukan sekumpulan pengguna atau tetangga (*neighbors*), yang memiliki penilaian serupa, lalu menggunakan algoritma tertentu untuk membuat prediksi atau top N rekomendasi untuk target pengguna[20]. Sedangkan sistem rekomendasi dengan pendekatan hybrid mengkombinasikan 2 tipe dari model-based dan memory-based. Pada penelitian ini pendekatan yang digunakan adalah memory based dimana seluruh perhitungan baik perhitungan kemiripan antar item maupun produk serta proses untuk menghasilkan rekomendasi dilakukan dalam memory secara online.

Dari algoritma yang terdapat dalam pendekatan memory-based, terdapat dua tipe yang berbeda yang dinamakan customer based collaborative filtering yang lebih jauh akan dibahas pada subbab 2.2.5.3

2.2.5.3 Algoritma Customer-Based CF

Algoritma customer-based collaborative filtering menggunakan teknik statistika untuk menemukan sekumpulan pengguna, dikenal sebagai tetangga (*neighbour*), yang memiliki catatan dalam riwayat transaksinya setuju atau memberikan penilaian yang serupa dengan pengguna yang menjadi sasaran. Setelah sekumpulan tetangga atau himpunan pengguna dengan karakteristik yang sama terbentuk, sistem menggunakan algoritma yang berbeda untuk menggabungkan kesukaan *neighbour* untuk menghasilkan prediksi atau rekomendasi N-teratas untuk pengguna yang menjadi target

Ilustrasi User Based CF



Gambar 2.6. Ilustrasi rekomendasi dengan metode customer based CF

2.2.6 Cosine Similarity

Untuk menghasilkan rekomendasi, perlu diketahui terlebih dahulu derajat kemiripannya dengan seluruh item/pengguna yang dibandingkan dalam dataset. Metode Cosine Similarity merupakan metode yang digunakan untuk menghitung similarity (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada vector space similarity measure[21]. Metode cosine similarity ini menghitung similarity antara dua buah objek (misalkan item A dan item B) yang dinyatakan dalam dua buah vector dengan menggunakan penilaian yang telah didapatkan dari masing-masing pengguna kepada tiap item. Untuk dua vektor A dan B dalam sebuah n ruang dimensional :

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \quad \text{F2.1. Rumus Cosine Similarity}$$

Similarity antara item A dan item B didapatkan dari hasil kali dua vektor dibagi dengan hasil kali panjang dua vektor. Cosine similarity memiliki kisaran antara -1 dan 1 dimana -1

menggambarkan korelasi yang sangat rendah dan 1 menunjukkan tingkat korelasi yang tinggi.

2.2.7 Euclidian Distance

Euclidean distance adalah perhitungan jarak dari 2 buah titik dalam Euclidean space. Euclidean space diperkenalkan oleh seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean ini biasanya diterapkan pada 2 dimensi dan 3 dimensi. Tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Euclidean Distance adalah metrik yang paling sering digunakan untuk menghitung kesamaan dua vektor. Rumus euclidean Distance adalah akar dari kuadrat perbedaan 2 vektor (root of square differences between 2 vectors)[22]:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

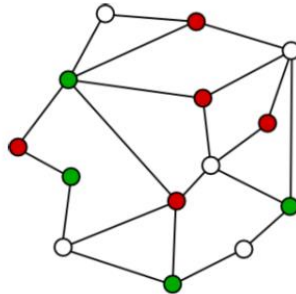
*F2.1. Rumus
Euclidian
Distance*

2.2.8 Basis Data Graf

Dari beberapa penelitian, ditemukan bahwa relational database masih memiliki beberapa kekurangan saat memproses data dengan dimensi yang besar dan dinamis seperti data yang digunakan pada jaringan sosial atau data yang digunakan untuk sistem rekomendasi karena seluruh data direpresentasikan dalam bentuk tabel, namun dengan adanya database graf seluruh data dapat direpresentasikan dengan sekumpulan *node* dan *edge* yang menggambarkan masing masing hubungan antar nodes [7].

Kunci untuk memahami basis data graf adalah dengan keunggulan yang sama dalam menyimpan baik data (pelanggan, produk) dan hubungan di antara mereka (yang membeli apa, siapa 'suka' produk apa, serta pembelian apa yang terjadi

pertama kali). Dalam basis data graf, kita tidak perlu lagi menggunakan model data semantik yang mahal dan terbatas, dengan proses ‘join’ yang tak terduga dari hubungan antar tabel. Sebaliknya, graph database mendukung banyak penamaan, hubungan langsung antara node dan entitas yang memberikan konteks semantik yang kaya untuk data



Gambar 2.7. Model basis data graf
(Sumber: Neo4j, 2012)[6]

Penelitian yang dilakukan oleh Betul J. Et al. Membuktikan bagaimana performa sistem rekomendasi yang lebih efisien dengan menggunakan database graf [23]. Selain itu, situs e-bay juga menggunakan neo4j dengan pemrosesan 10-100 juta lebih cepat dari pemrosesan SQL[24].

Diambil dari buku Neo4j in Action, dengan topik mencari relasi pertemanan dengan percobaan performa antara mysql dan relational database dengan percobaan pada 1 juta orang/objek, perbandingan performa antara kedua basis data graf ditampilkan dalam tabel berikut dengan beberapa perbedaan layer dan tabel :

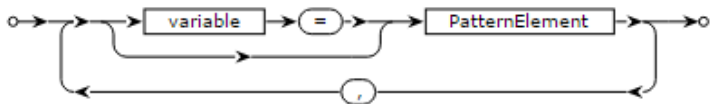
depth	neo4j	RDBMS
1
2	0.01	0.016
3	0.168	30.267
4	1.359	1543.505
5	2.132	unfinished

2.2.8.1 Neo4j

Neo4j merupakan basis data NoSQL dengan sistem graf. Neo4j menyimpan relasi antar objek dalam struktur seperti graf, dimana setiap objek merujuk ke objek lainnya secara langsung. Dalam menambahkan maupun mengambil data, Neo4j memiliki bahasa sendiri yang disebut Cypher. Neo4j ditulis dalam bahasa Java dengan lisensi GPL V3 oleh Neo Technologies artinya Neo4j termasuk basis data yang open source.

2.2.8.2 Cypher Query Language (CQL)

Cypher merupakan bahasa pemrosesan query dari database neo4j. Cypher merupakan bahasa analitik yang memungkinkan developer untuk membaca dan menuliskan perintah ke dalam database graf. Salah satu kelebihan dari bahasa ini adalah mudah dimengerti oleh bahasa manusia. Fitur baru dari neo4j kini telah mendukung adanya pelabelan dan index sehingga kini setiap nodes dari graf dapat diberi label dan property dari setiap nodes dapat digunakan sebagai index [7]. Node dan ekspresi relationship adalah blok untuk pola yang lebih kompleks.



Gambar 2.8. Struktur Cypher Query Language

Pola dapat ditulis terus menerus atau dipisahkan dengan koma. Kita dapat merujuk ke variabel yang dideklarasikan sebelumnya atau menggunakan yang baru. Berikut ini merupakan struktur pemrosesan *Cypher Query Language*.

2.2.9 Node.js

Node.js merupakan sebuah salah satu perangkat lunak open source yang didisain untuk mengembangkan aplikasi berbasis web dan dibangun pada Google Chrome's JavaScript Engine (V8 Engine). Node.js diciptakan oleh Ryan Dahl pada tahun 2009. Node.js dieksekusi sebagai aplikasi server. Platform ini menggunakan bahasa pemrograman javascript dan menggunakan teknik non-bloking untuk mempercepat proses. Teknik non-bloking adalah sebuah metode penyelesaian sebuah fungsi. Teknik non-bloking yaitu teknik dimana node.js akan melakukan eksekusi secara independen. Node.js akan mengeksekusi sebuah operasi tanpa harus menunggu operasi sebelumnya selesai dieksekusi, sehingga menghasilkan aplikasi web yang lebih cepat dan efisien [25]. Beberapa keunggulan node.js adalah:

- Javascript secara native mendukung JSON yang merupakan standar transfer data yang banyak dipakai saat ini sehingga untuk mengkonsumsi data-data dari pihak ketiga pemrosesan di Node.js akan sangat mudah sekali.
- Database NoSQL mendukung langsung Javascript sehingga interfacing dengan database ini akan jauh lebih mudah.
- Penulisan web application dapat dilakukan pada satu bahasa, yang mengurangi context switch antara client dan server.
- JavaScript adalah target kompilasi dan terdapat banyak bahasa yang telah siap dikompilasi dengan JavaScript
- Node menggunakan virtual machine V8 yang mengikuti standar ECMAScript, dengan kata lain tidak perlu menunggu browser untuk siap menggunakan fitur JavaScript baru di Node.js

2.2.9.1 Express JS

Express JS merupakan framework aplikasi web Node.JS yang fleksibel dan minimal, serta menyediakan fitur yang kuat untuk web dan mobile. Express JS adalah sebuah modul framework

pada Node.JS untuk mempermudah menyediakan HTML web pada file tersendiri dengan format “.ejs”. Express JS memberikan kemudahan dengan adanya fitur untuk mengirimkan data menuju file HTML dari Node.JS. Data variabel yang dikirimkan pada file Express JS dapat diakses dengan menuliskan kode “<%=variabel%>”. Data yang dikirimkan dapat berupa array, yang dipanggil melalui looping pada kode HTML. File Express JS dapat diakses oleh Node.JS melalui pemanggilan fungsi render diikuti dengan parameter lokasi file, dan variabel yang dikirimkan menuju HTML[26].

2.2.9.2 NPM

NPM merupakan singkatan dari *Node Package Manager*, yang merupakan sebuah pengatur paket (package manager) untuk Node.JS yang dibuat pada tahun 2009. NPM pada awalnya dibuat sebagai proyek *open-source*, memiliki tujuan untuk mempermudah pengembang JavaScript dalam menggunakan dan membagikan modul-modul kode. Kemudian didirikannya NPM, Inc. pada tahun 2014 oleh penemunya Isaac Z. Schlueter, dengan Laurie Voss dan Rod Boothby. NPM merupakan sekumpulan paket kode open-source yang terdiri dari modul seperti Node.JS, aplikasi web pada sisi klien (*client-side*), aplikasi mobile, robot, router, dan berbagai kebutuhan komunitas JavaScript lainnya. NPM dapat digunakan dengan cara mengetikkan perintah “npm install” pada *Command Line Interface*, yang diikuti dengan nama modul yang ingin digunakan[26]. NPM sendiri telah menyediakan package khusus untuk pemrosesan menggunakan neo4j.

2.2.9.3 Forbento Dataset

Forbento adalah bidang usaha yang bergerak di bidang *e-commerce* dengan melakukan penjualan beraneka ragam bento tools. Proses transaksi forbento dilakukan secara online dan offline. Penjualan online forbento bisa dilakukan melalui berbagai channel yang berbeda seperti website forbento, email, tokopedia, bukalapak, BBM, maupun whatsapp. Seluruh data

penjualan dimasukkan dalam sebuah basis data mySQL untuk memudahkan pemantauan transaksi. Dataset forbento yang digunakan terdiri dari tiga komponen yaitu data pelanggan, data produk dan catatan transaksi yang terjadi saat pelanggan membeli produk.

2.2.10 Iterative and Incremental Development

Ide dasar dari metode pengembangan ini ialah untuk mengembangkan sebuah sistem melalui siklus berulang (iterative) dan dalam porsi yang lebih kecil pada suatu waktu (incremental), memungkinkan pengembang perangkat lunak untuk mengambil keuntungan dari apa yang telah dipelajari selama pengembangan bagian-bagian awal versi sistem. [27]

Inception

Inception merupakan tahap permulaan untuk mengidentifikasi pengembangan sistem. Pada tahap ini dilakukan beberapa aktivitas diantaranya adalah menentukan ruang lingkup dari penelitian, proses penggalan dan analisis kebutuhan pengguna, proses perancangan awal perangkat lunak, dan permodelan UML yang melingkupi berbagai diagram proses seperti domain model, use case diagram, activity diagram & sequence diagram.

Elaboration

Pada fase elaboration yang merupakan fase perluasan atau perencanaan. Pada tahap ini dilakukan desain secara lengkap berdasarkan hasil analisis di tahap inception. Aktivitas terkait yang ada di tahap ini antara lain adalah pembuatan desain arsitektur sistem (*architecture pattern*), desain format data/struktur data, desain database sistem, tampilan sistem (*user interface*), serta menentukan *design pattern* yang digunakan.

Construction

Dalam tahapan konstruksi, dilakukan proses implementasi hasil desain dan proses pengujian hasil implementasi. Aktivitas yang ada dalam tahap ini diantaranya adalah pengujian hasil analisis dan desain, penentuan coding program yang digunakan, pembuatan program, dan optimasi program.

Transition

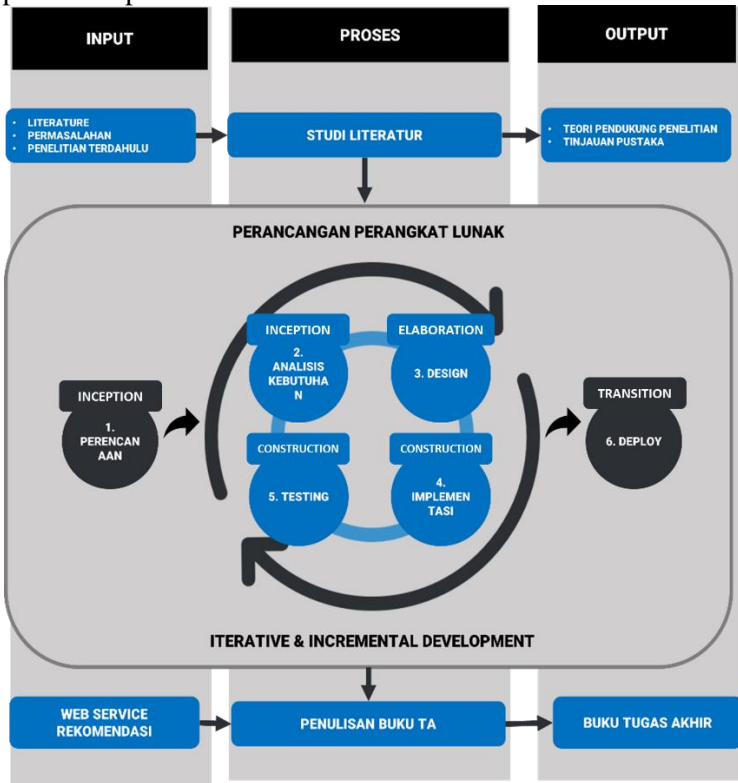
Transition (transisi), merupakan tahap untuk menyerahkan sistem aplikasi ke pelanggan (*roll-out*), yang umumnya mencakup pelaksanaan pelatihan kepada pengguna dan testing beta aplikasi terhadap ekspektasi pengguna.

Halaman ini sengaja dikosongkan

BAB III

METODOLOGI PENELITIAN

Pada bab metodologi penelitian akan menjelaskan alur dari penelitian berikut ini beserta penjelasan dari setiap tahapan dalam penelitian. Berikut ini ialah diagram alur dari pengerjaan penelitian pada Gambar 3.1



Gambar 3.1. Alur metodologi Penelitian

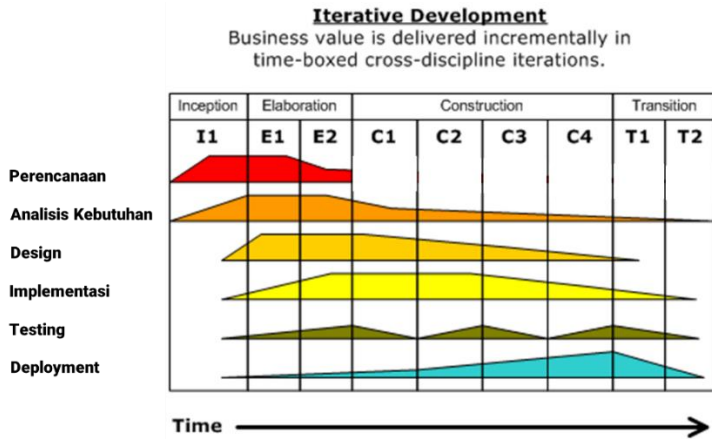
3.1 Metode Pengerjaan

3.1.1 Studi Literatur

Tahapan awal pengerjaan Tugas Akhir ini adalah studi literatur. Studi literatur digunakan untuk menentukan topik dengan cara menganalisa latar belakang permasalahan terkait sistem rekomendasi pada *e-commerce* berikut metodenya serta menentukan rumusan permasalahan yang diangkat. Pada tahapan ini pun juga ditentukan batasan masalah, tujuan, serta manfaat penentuan topik penelitian. Untuk mendukung latar belakang permasalahan dan rumusan masalah yang sesuai dengan topik, dilakukan peninjauan pustaka mulai dari studi penelitian sebelumnya terkait dengan implementasi sistem rekomendasi pada *e-commerce*, penggunaan metode *collaborative filtering* dengan pendekatan *customer based*, permodelan database graf neo4j. Adapun literatur yang digunakan berasal dari jurnal ilmiah nasional maupun internasional, makalah penelitian, *e-book*, maupun buku-buku yang dapat dijadikan bahan acuan pengerjaan tugas akhir ini.

3.1.2 Perancangan Perangkat Lunak

Perancangan web service sistem rekomendasi menggunakan metode *iterative & increment development* yang terdiri dari 4 fase dan 6 proses di dalamnya yang akan dijabarkan dalam uraian berikut:



Gambar 3.2. Fase Iterative Development

3.1.2.1 Perencanaan (Fase *Inception*)

Pada tahap ini dilakukan analisis dan perencanaan sistem yang ada dalam web service, menentukan protokol yang digunakan, memilih implementasi teknologi yang tepat untuk melakukan komunikasi antar aplikasi serta menentukan desain pattern yang tepat untuk web service yang akan dibuat. Selain itu dilakukan analisis terhadap cara kerja penggunaan API oleh pengguna.

Pada penelitian ini, dirancang desain arsitektur web service yang akan menjadi panduan dalam pembangunan sistem rekomendasi. Pada layer aplikasi digunakan teknologi REST yang menggunakan protokol HTTP untuk berkomunikasi antar aplikasi sedangkan format data yang digunakan adalah format JSON yang mudah dibaca oleh mesin ataupun manusia.

3.1.2.2 Analisis Kebutuhan (Fase *Inception*)

Proses selanjutnya pada fase *inception* adalah analisis kebutuhan yang dilakukan untuk mendefinisikan kebutuhan yang harus ada di dalam sistem yaitu terkait kebutuhan fungsional sistem dan analisis fitur pada sistem rekomendasi

yang dihasilkan. Berikut ini adalah daftar kebutuhan fungsional yang akan dirancang:

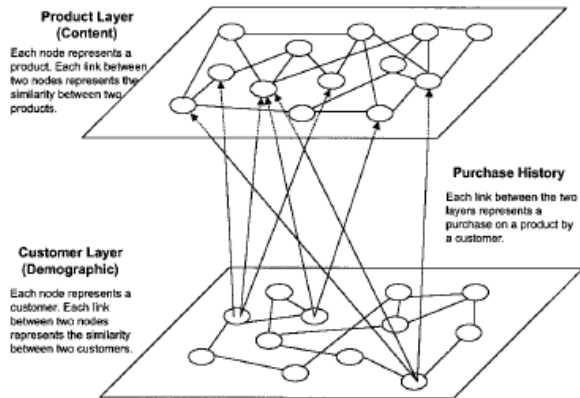
- Aplikasi *client* dapat mengirimkan data dalam bentuk JSON ke aplikasi *server*
- Aplikasi *client* dapat meminta dan menerima hasil rekomendasi dari *server*
- Aplikasi *client* dapat menampilkan hasil rekomendasi ke dalam *e-commerce*
- Aplikasi *server* dapat menerima data yang dikirimkan
- Aplikasi *server* dapat memodelkan data dalam bentuk graph
- Aplikasi *server* dapat menghitung kemiripan antar pelanggan dan produk
- Aplikasi *server* dapat menghasilkan rekomendasi dari algoritma collaborative filtering.
- Aplikasi *server* dapat mengupdate data saat ada permintaan dari client

3.1.2.3 Design

Berdasarkan hasil perencanaan pada tahap sebelumnya akan digunakan untuk membuat gambaran desain dari fitur dan fungsionalitas terhadap aplikasi yang akan dibangun nantinya. Pada tahap ini akan dibuat beberapa desain antara lain:

a. Desain Database (Model Basis Data graph)

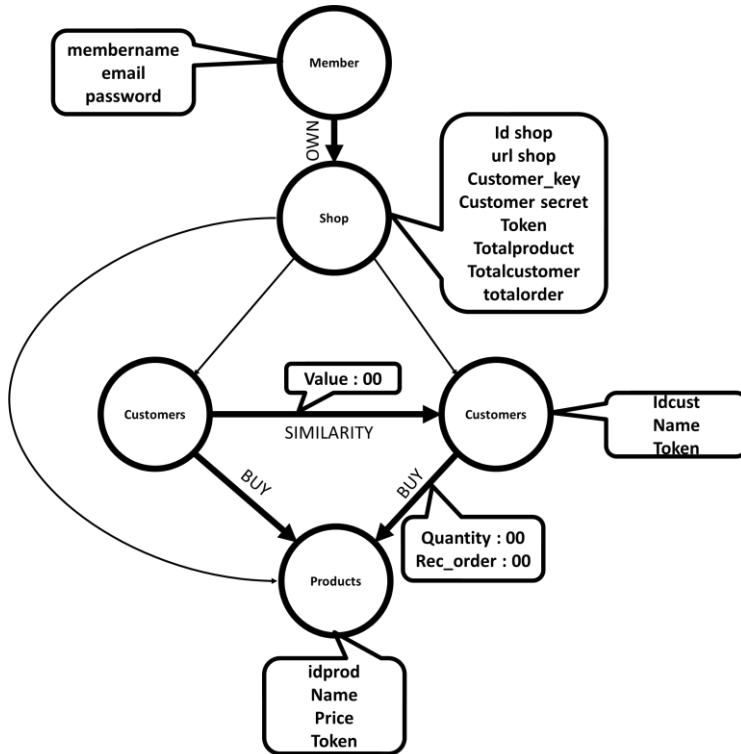
Permodelan data yang digunakan dalam penelitian ini merujuk pada model 2 layer database graf yang dirujuk dari penelitian Zan Chung Huan yang merepresentasikan model antara produk, pelanggan dan transaksi yang terjadi diantara keduanya seperti gambar 3.3 berikut:



Gambar 3.3. Model 2 Layer Graph Database

Sumber: (Huan, Zang, 2004) [28]

Pada desain model data graf tersebut, dapat merepresentasikan hubungan antara setiap produk yang ada di dalam suatu bisnis beserta pelanggannya melalui hubungan yang ada dari catatan transaksi antara keduanya. Seluruh data digambarkan dalam node dan edge pada gambar 3.3. berikut



Gambar 3.4. Permodelan data rekomendasi

Tabel 3.1. Daftar node pada graf

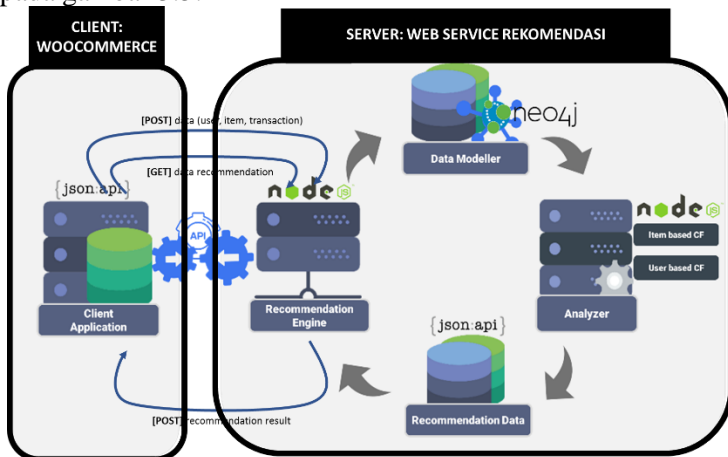
Node Label	Properties
Pelanggan	<ul style="list-style-type: none"> • Id pelanggan • Nama pelanggan
Produk	<ul style="list-style-type: none"> • Id produk • Nama produk • Harga produk
Relationship Label	Properties
Penilaian	<ul style="list-style-type: none"> • Id penilaian • Nilai

Tabel 3.2. Daftar Relationship pada graf

Relationship Label	Properties
Penilaian	<ul style="list-style-type: none"> • Id penilaian • Nilai
Pelanggan_Similarity	<ul style="list-style-type: none"> • Id sim • Similarity_value

b. Desain Sistem

Desain arsitektur web service yang akan dibuat dapat dilihat pada gambar 3.5.

**Gambar 3.5. Desain Infrastruktur Web Service Rekomendasi**

1. Client Application

Penjelasan:

Platform e-commerce berbasis woocommerce yang digunakan oleh forbento untuk menampilkan hasil rekomendasi kepada pelanggan

Bagian yang dikerjakan:

- Pembuatan Plugin rekomendasi
- Pembuatan plugin bulk import order secara otomatis
- Pembuatan plugin bulk import pelanggan otomatis
- Kustomisasi backend & frontend tema woocommerce

Bagian yang telah tersedia: Package woocommerce dan wordpress.
2. Recommendation Engine
Penjelasan : Web service yang digunakan pengguna untuk melakukan management data toko.
Bagian yang dikerjakan: <ul style="list-style-type: none"> • Pembuatan halaman dashboard • Pembuatan fungsi update data otomatis melalui dashboard, yaitu update pelanggan, update produk dan update order ke database graf. • Pembuatan halaman register member & fungsinya • Pembuatan halaman login member dan fungsinya • Pembuatan halaman register toko dan fungsinya • Pembuatan halaman login toko & fungsinya • Pembuatan halaman profil pengguna • Pembuatan halaman data pelanggan dan data produk
Bagian yang telah tersedia: -
3. Data Modeller
Penjelasan: Data Handler yang ada di dalam web service dan berfungsi untuk mengambil data dari woocommerce API dan memodelkannya secara otomatis ke dalam graf neo4j
Bagian yang dikerjakan: <ul style="list-style-type: none"> • Pembuatan fungsi convert data di dalam web service untuk mengambil dan mengubah order, data pelanggan dan data produk dari woocommerce API dalam format json. • Pembuatan fungsi <i>data modeller</i> dalam web service untuk mengubah data json menjadi graf sesuai dengan model yang telah direncanakan
Bagian yang sudah tersedia: -Browser neo4j
4. Analyzer/ Pemroses Rekomendasi

Penjelasan:

Bagian dari web service yang berfungsi untuk memproses rekomendasi dan mengimplementasikan algoritma collaborative filtering

Bagian yang dikerjakan:

- Pembuatan fungsi analyzer dengan penerapan algoritma collaborative filtering untuk mengambil rekomendasi produk dari user yang memiliki kemiripan tertinggi.
- Pembuatan fungsi untuk penerapan metode cosine similarity dan euclidian distance untuk menghitung kemiripan pelanggan

Secara umum aplikasi yang akan dibuat terdiri dari dua bagian besar yaitu aplikasi yang ada pada bagian client dan aplikasi yang ada pada bagian server. Aplikasi *client* digunakan untuk mengumpulkan data yang diperlukan untuk rekomendasi dan meminta hasil rekomendasi dari server. Aplikasi yang ada di server bertugas untuk mengolah data untuk memberikan hasil rekomendasi yang siap dikirim ke client dalam format JSON.

Alur penjelasan dari arsitektur di atas dimulai dari API yang berada pada sisi client mengumpulkan data yang diperlukan (data produk, data pelanggan, data transaksi, serta data pelanggan yang sedang aktif) dari *e-commerce* yang digunakan untuk diubah menjadi JSON agar dapat berkomunikasi dengan server. Komunikasi client ke server dilakukan melalui protokol HTTP dengan memanggil URL tertentu yang telah memiliki token tersendiri untuk mengakses web service.

Aplikasi server menggunakan nodejs untuk memproses permintaan rekomendasi dan mengolah data yang diterima. Setelah server sukses menerima data maka sistem akan melakukan pembacaan & permodelan data ke model graph serta mengeksekusi fungsi fungsi dan perintah untuk mengimplementasikan algoritma collaborative filtering (customer based CF) di basis data neo4j yang terletak pada server melalui analyzer yang dijalankan melalui sistem node js.

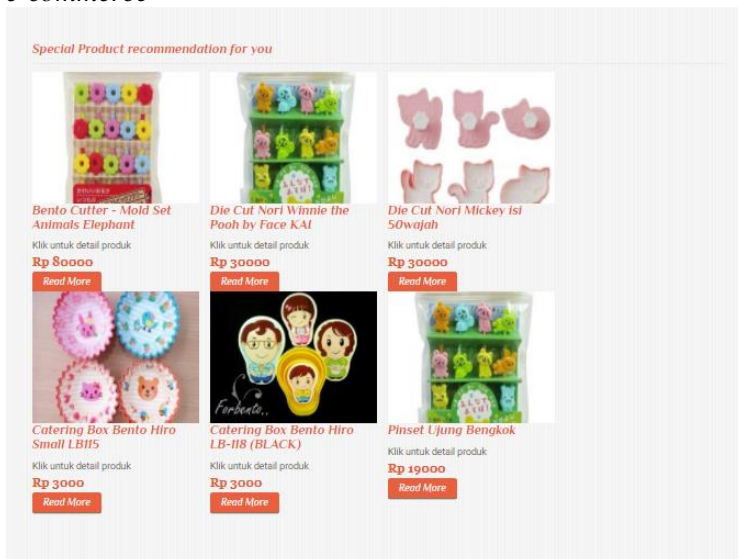
Selanjutnya hasil query data diproses oleh web service dan hasilnya dikirim dalam format objek JSON yang siap digunakan oleh aplikasi pengguna. Basis data graf dalam web server hanya akan menyimpan data yang aktif dalam 3 jam terakhir untuk meminimalisir overload data.

Adapun fungsi fungsi rest yang akan diimplementasikan dalam web service ini adalah fungsi get dan post dalam berbagai fitur seperti menambah data, melihat relasi graph, melihat data pelanggan & produk, melihat aktifitas pembelian pelanggan tertentu serta mendapat rekomendasi

c. Desain User Interface

Berikut ini merupakan gagasan *user interface* dari sistem yang akan diimplementasikan :

Gambar 3.2 merupakan rancangan tampilan hasil implementasi penggunaan web service berupa panel rekomendasi di halaman *e-commerce*



Gambar 3.6. Perancangan desain UI hasil implementasi
Sumber: Woocommerce, 2017

3.1.2.4 Implementasi (Fase Construction)

Setelah melalui tahap desain, dilakukan pengkodean terhadap web service rekomendasi yang menggunakan bahasa pemrograman HTML, javascript, dan cypher.

1. Pengkodean aplikasi klien

Dalam tahap ini dilakukan pengkodean beberapa fungsi untuk mengumpulkan data yang dibutuhkan dari aplikasi klien dengan membuat modul tambahan dari simulasi data forbento yang telah disiapkan. Aplikasi client akan mengubah data dalam format JSON dan melakukan generate token untuk mengakses web service.

2. Pengkodean aplikasi server

Selanjutnya dibuat aplikasi server untuk merespon permintaan yang diberikan oleh client. Pembangunan aplikasi server dilakukan dengan menggunakan node js untuk memodelkan data yang diterima dalam bentuk basis data graf sebelum dilakukan pemrosesan dengan analyzer.

3. Pengkodean analyzer

Selanjutnya dilakukan pembuatan analyzer yang berfungsi sebagai pemrosesan dalam implementasi algoritma collaborative filtering. Adapun proses penerapan algoritma collaborative filtering dijelaskan dalam uraian berikut:

3.1. Menghitung similarity

Setelah kita mendapatkan model dari database graf, kita dapat melakukan perhitungan similarity antar produk dan pelanggan dengan menggunakan rumus cosine similarity antara satu pelanggan dengan pelanggan lainnya. Perhitungan similarity antar pelanggan dan produk dilakukan melalui cypher query pada node.js yang akan menampilkan hasil dalam memory sehingga dapat langsung dibaca oleh server neo4j. Hasil similarity antar user kemudian dimodelkan menjadi sebuah relationship baru pada graph yang akan di update.

3.2. Menghitung prediksi penilaian

Setelah kita mendapatkan nilai kemiripan antar pelanggan dan produk, kita dapat melakukan prediksi pada barang barang yang

belum dinilai oleh target pelanggan yang menginginkan rekomendasi. Proses perhitungan prediksi penilaian dilakukan dengan mencari terlebih dahulu pelanggan yang paling mirip dengan target pelanggan dari proses sebelumnya, lalu dengan menggunakan algoritma k-nearest neighbors didapatkan kelompok pelanggan dengan kesamaan tertinggi untuk kemudian dilakukan prediksi penilaian ke produk tertentu berdasarkan penilaian yang diberikan oleh pelanggan lain yang merupakan neighborhood dari target pelanggan.

3.3. Generate rekomendasi

Setelah nilai prediksi diketahui, maka dapat ditentukan N rekomendasi terbaik berdasarkan nilai yang diperoleh. Perolehan rekomendasi tergantung pada algoritma yang digunakan. Pada algoritma customer based, rekomendasi didapatkan berdasarkan kemiripan antar pelanggan

3.4. Membaca hasil rekomendasi

Hasil rekomendasi yang ada pada server Neo4j harus dapat dibaca melalui web server node.js. Oleh karena itu dibuatlah sebuah REST API yang dapat membaca hasil query dari server neo4j. Data yang didapatkan berupa data JSON sehingga dapat dibaca oleh aplikasi klien.

3.1.2.5 Testing (Fase Construction)

Pengujian Pada tahap ini dilakukan pengujian aplikasi untuk memastikan aplikasi dapat berjalan sesuai dengan yang diharapkan oleh pengguna dan mencatat semua bug dan error yang ada pada web service. Dalam tahap pengujian dilakukan percobaan implementasi API pada salah satu website ecommerce. Dalam hal ini platform yang digunakan untuk melakukan pengujian adalah Platform woocommerce yang digunakan oleh Forbento Beberapa pengujian yang dilakukan meliputi:

- Uji fungsional sistem

- Uji akurasi rekomendasi

Uji Fungsionalitas dilakukan untuk memastikan bahwa seluruh kebutuhan fungsional telah terpenuhi, sedangkan uji akurasi rekomendasi dilakukan untuk memastikan hasil rekomendasi yang dihasilkan benar dan dapat diterima.

3.1.2.6 Deployment (Fase Transition)

Dalam proses ini dilakukan proses deployment dengan pembuatan dokumentasi API yang siap diberikan kepada pengguna.

3.1.3 Penulisan buku tugas Akhir

Penulisan Buku Tugas Akhir adalah tahap yang menghasilkan buku Tugas Akhir. Penulisan buku ini dilakukan bersamaan dengan tahapan penelitian yang lainnya. Harapannya, luaran berupa buku tugas akhir dapat dijadikan sebagai referensi untuk penelitian selanjutnya.

Halaman ini sengaja dikosongkan

BAB IV

PERANCANGAN

Pada bab ini, akan dijelaskan tentang rancangan dari tugas akhir yang terdiri dari subjek dan objek dari tugas akhir ini. Selain itu akan dijelaskan juga proses pemilihan subjek dan objek dari tugas akhir yang akan dilakukan. Pada bab ini akan dijelaskan mengenai tahapan – tahapan yang termasuk ke dalam perancangan prototype dan implementasi web service. Tahapan – tahapan tersebut meliputi Pengumpulan data, Analisis kebutuhan pada client dan server, analisis perancangan fitur, analisis product backlog hingga protokol pengujian aplikasi. Setiap tahapan yang dilakukan menggunakan pendekatan Iterative dan Incremental Development. Pada tahapan perencanaan yang terdapat dalam laporan ini telah melalui tahapan iterasi sebanyak 3 kali iterasi perancangan yang masing-masing terdiri dari perancangan data, perancangan aplikasi client dan perancangan aplikasi server. Berikut adalah penjelasan dari setiap tahapan – tahapan perancangan.

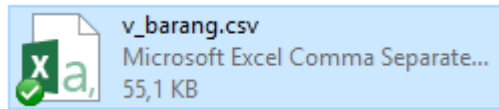
4.1 Perancangan Data

4.1.1 Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang berasal dari toko online alat memasak Forbento yang terdiri dari 3 data penjualan yaitu data seluruh produk, data seluruh customer dan data penjualan terakhir. Data yang didapatkan berupa 3 buah file dengan format .csv yaitu sebagai berikut

a. Data Produk

Data produk yang didapatkan berupa daftar seluruh produk yang ada di dalam katalog forbento dalam seluruh kategori. Data ini disajikan dalam format .csv untuk kemudian diolah dalam database.



Gambar 4.1. File csv data produk

Data Produk yang didapatkan terdiri dari 1137 baris dan terdiri dari 3 atribut yaitu id produk, nama produk, dan harga produk. Adapun id produk merupakan identifikasi dari katalog website forbento, nama produk dari katalog website forbento, dan harga produk terbaru dari forbento dengan satuan rupiah. Contoh data ditampilkan dalam Gambar 4.2 berikut:

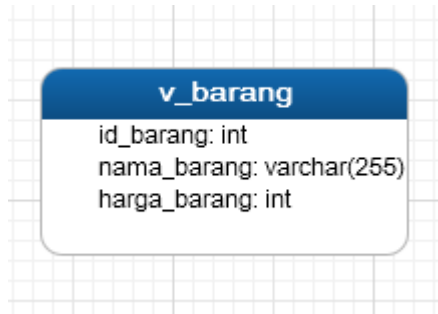
```

1  "id","nama","harga"
2  "940","Die Cut Nori Face KAI isi 50wajah","30000"
3  "124","Sumpit Melamin","26000"
4  "125","Lunch Pick - Lovely flower heart ribbon star","29000"
5  "7","Lunch Pick - Melody","29000"
6  "8","Lunch Picks - Smileys","30000"
7  "86","Lunch Flag - Vehicle","29000"
8  "87","Lunch Flag - Animal","29000"
9  "88","Pop Flower Pick","30000"
10 "89","Lunch Pick - Pop Round","35000"

```

Gambar 4.2. Contoh data produk dalam format csv

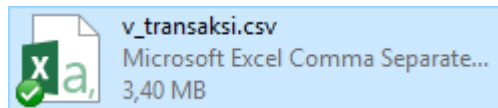
Seluruh data tersebut diolah dan diimport ke dalam database dengan tipe data integer untuk id_barang, tipe data varchar dengan length maksimum untuk nama_barang, dan tipe data integer untuk atribut harga_barang seperti tampilan dalam Gambar 4.3 berikut.



Gambar 4.3. Tabel produk dalam database

b. Data Pengiriman

Data pengiriman merupakan data yang berisi tentang catatan penjualan forbento secara detail yang menggambarkan informasi pengiriman barang untuk customer yang melakukan transaksi dan juga berisi informasi setiap pelanggan. Data ini diberikan dalam format .csv yang berisi 86.881 baris transaksi



Gambar 4.4. File csv data pengiriman

Data transaksi memiliki 10 atribut sebagai berikut:

- id: menunjukkan nomor identifikasi transaksi
- tanggal : tanggal transaksi / checkout barang
- nama : nama pelanggan yang melakukan transaksi
- alamat: alamat pengiriman barang / alamat pelanggan yang dituju
- telp: nomor telepon pelanggan
- email: email pelanggan untuk konfirmasi
- total pembelian : total harga yang harus dibayar oleh pelanggan
- detail: detail barang yang dibeli oleh pelanggan dimana setiap item memiliki informasi nama produk, warna

produk, ukuran produk, harga satuan, jumlah item yang dibeli dan total harga pada item tersebut.

- Status: menunjukkan status pengiriman sudah dilakukan atau belum selesai

```

1 ["id","tanggal","nama","alamat","kota","telp","email","total_pembelian","detail","status"
2 "1","2011-04-13","ayu","ayu","1234","ayu.fatmawati@gmail.com","30000","Anda mendapat pesanan order sebagai berikut :
3 1 Food Cup - isi 3 bentuk bulat() warna = ,ukuran = @ 30000 = 30000
4 Total Biaya = Rp. 30000","Selesai"
5 "2","2011-04-28","ritzki marika","Perum. BJI Kamp. Crewed jl. ceria 5 blok f44 no. 20 RT 10 / 13. Bekasi Timur. 17111",
6 ,"085693519040","ritzki_marika@yahoo.com","8000","Anda mendapat pesanan order sebagai berikut :
7 1 Nori - Rumput Laut Kering - Small Nori - Rumput Laut Kering - Small() warna = ,ukuran = @ 8000 = 8000
8 Total Biaya = Rp. 8000
9 ","Selesai"
10 "3","2011-05-05","Novy F. Santoso","Perumahan Alam Sutera
11 Jl. Sutera Feronia 4 No. 56
12 Serpong, Tangerang","081315962175","novy301175@gmail.com","231000","Anda mendapat pesanan order sebagai berikut :

```

Gambar 4.5. Contoh data pengiriman dalam format csv

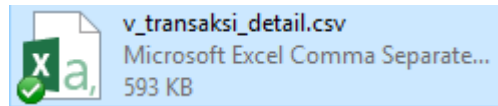
Seluruh data tersebut diolah dan diimport ke dalam database dengan tipe data yang sesuai yaitu tipe data integer untuk id_transaksi, tipe data date untuk tanggal transaksi, tipe data text untuk alamat pelanggan dan detail pembelian, tipe data varchar untuk nama pelanggan, kota, telepon, email dan status sehingga menjadi seperti tampilan pada Gambar 4.6:

v_transaksi	
id_transaksi:	int
tanggal:	date
nama_pelanggan:	varchar(255)
alamat:	text
kota:	varchar(255)
telp:	varchar(20)
email:	varchar(50)
total_pembelian:	int
detail:	text
status:	varchar(255)

Gambar 4.6. Tabel pengiriman dalam database

c. Data Transaksi

Data transaksi merupakan data yang berisi tentang rangkuman catatan transaksi forbento secara ringkas yang menggambarkan hubungan antara transaksi dan setiap barang yang dibeli oleh customer dalam seluruh transaksi yang telah dilakukan tanpa data pelanggan. Data ini diberikan dalam format .csv yang berisi 23.639 baris keranjang belanja.



Gambar 4.7. File csv data transaksi

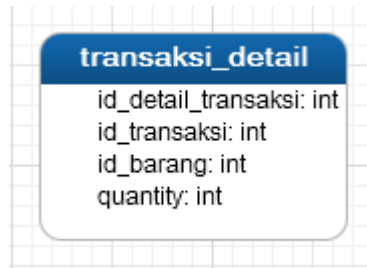
Data keranjang transaksi memiliki 4 atribut diantaranya adalah sebagai berikut:

- id: nomor identifikasi informasi produk & transaksi
- id_transaksi: nomor identifikasi transaksi
- product_id: nomor identifikasi produk
- quantity: jumlah produk yang dibeli

1	"id","id_transaksi","product_id","quantity"
2	"1","946","552","1"
3	"2","946","460","1"
4	"3","946","513","1"
5	"4","946","288","1"
6	"5","946","489","1"
7	"6","946","529","1"
8	"7","947","472","1"
9	"8","948","531","1"
10	"9","948","542","1"
11	"10","948","483","1"
12	"11","948","25","1"
13	"12","950","288","1"
14	"13","950","539","1"
15	"14","950","408","1"

Gambar 4.8. Contoh data transaksi dalam format csv

Seluruh data tersebut diolah dan diimport ke dalam database dengan tipe data yang sesuai yaitu tipe data integer untuk id_detail_transaksi/id, id_transaksi, id_barang/product_id dan quantity.



Gambar 4.9. Tabel pengiriman dalam database

4.1.2 Pemilihan Atribut/Analisis kebutuhan data

Dalam menghasilkan rekomendasi, dilakukan analisis kebutuhan data untuk mengetahui data yang dibutuhkan untuk menghasilkan rekomendasi yang sesuai. Dengan data yang telah dilakukan dilakukan pemilihan atribut kunci minimal yang dapat digunakan untuk memproses sebuah rekomendasi yaitu sebagai berikut:

Data Produk

Data produk dari data Gambar 4.8 dapat diambil dari tabel v_barang dalam database yang telah memiliki nama dan harga barang di dalamnya. Data ini nantinya akan digunakan sebagai objek rekomendasi yang akan diberikan pada setiap pelanggan. Dalam hal ini data product haruslah unik dan minimal memiliki nomor identifikasi(id produk)

Data Pelanggan

Untuk mendapatkan data pelanggan dapat diambil dari data pengiriman pada tabel v_transaksi untuk mengetahui kepada siapa rekomendasi tersebut akan diberikan. Data pelanggan yang diperlukan harus unik dan minimal memiliki nomor identifikasi (id pelanggan)

Data Transaksi (Nilai penjualan)

Data transaksi yang dibutuhkan adalah dalam transaksi tertentu, perlu diketahui siapa yang melakukan pembelian dan apa yang dibeli. Data ini didapatkan dari perpaduan antara 2 tabel yaitu *v_transaksi* dan *v_detail_transaksi* sehingga harus dilakukan pra pemrosesan data pada langkah selanjutnya.

Dalam hal ini data penjualan digunakan untuk menggantikan data penilaian yang biasa digunakan dalam sistem rekomendasi pada umumnya menghimpun data rating produk yang dinilai oleh sekumpulan orang tertentu. Untuk menghasilkan nilai penjualan yang akurat dan cukup baik untuk menggantikan asumsi penilaian user terhadap suatu barang, maka digunakan *reccurence order* atau pemesanan berulang. Adanya perulangan pembelian pada transaksi yang berbeda menunjukkan kecondongan pelanggan pada produk tersebut.

4.1.3 Pra-pemrosesan Data

Sebelum diproses, data harus terlebih dahulu mengalami proses persiapan data atau pra-pemrosesan. Data pra pemrosesan menunjukkan tipe-tipe proses yang menggunakan data mentah untuk ditransformasi ke suatu format yang lebih mudah dan efektif untuk kebutuhan rekomendasi agar dapat diolah dengan baik saat penerapan algoritma pada langkah selanjutnya.

4.1.3.1 Pra-proses data Pelanggan

a. Pemilihan atribut

Dari data yang telah didapatkan, data customer bercampur menjadi satu dalam data transaksi, sehingga harus dilakukan pemilihan data. Untuk mendapatkan data customer yang tercatat terdapat beberapa langkah yang harus dilakukan yaitu dengan melakukan penyaringan data pelanggan pada tabel

pengiriman (v_transaksi) atribut terkait data customer yang bisa didapatkan dengan query di bawah ini

1. **SELECT** nama_pelanggan, alamat, kota, telp, email
2. **FROM** v_transaksi

Kode 4.1. Query untuk filter informasi pelanggan

Dari query pada Kode 4.1 berhasil dilakukan penyaringan data sehingga didapatkan data pelanggan dengan jumlah 16365 baris. Hasil eksekusi query dapat dilihat pada Gambar 4.10

nama_pelanggan	alamat	kota	telp	email
Putri Dianita Paidi	Jalan Karah	Surabaya	0857 3049 2830	forbentoarie@gmail.com
Yani Dwi handayani	Jl. H. Tekel F	Bekasi	+62 813-1002-0161	Forbentobuayu@gmail.com
I Gusti Ayu Ardhi Paryanthi	Jalan Salak	Bali	085 338 576 569	forbentoarie@gmail.com
Ibu Windri Widyaniingsih	Karaba Inda	Karawang	0857 2612 2233	forbentoarie@gmail.com
Bayu Kristanti	Jalan Elang	Tangeran	0818742298	green_genbu@yahoo.com
Ibu Hartatik SE	BRI Unit Rer	Denpasar	081999301011	forbentotokopedia@gmail.com
putri indah sari	Panglima St	(Null)	082232609096	sunas@sunasdyana.com
Bunda Viya	Jalan Raya T	Malang	081 933 130 100	forbentoarie@gmail.com
Rini Purnawati Widodo	Perum Saur	Karawang	082111490719	rini.widodo888@gmail.com
Juliana Agustin/ Widyawati	Semolowari	Surabaya	08179314645	fransiska_widyawati@yahoo.com

Gambar 4.10. Hasil penyaringan data

b. Pembersihan data

Data-data yang telah terkumpul selanjutnya akan mengalami proses pembersihan. Proses pembersihan data dilakukan untuk membuang/membenarkan baris yang tidak valid, menstandarkan attribut-attribut, dan merasionalisasi struktur data. Pembersihan data perlu dilakukan dari kumpulan data yang didapatkan karena masih terdapat duplikasi data dari proses sebelumnya.

Pertama-tama dilakukan proses pembersihan data pada data pelanggan. Pada tahap ini data pelanggan tidak memiliki nomor identifikasi yang unik dan masih memiliki isi atribut yang tidak seragam.

Berikut ini merupakan contoh data pelanggan yang masih belum dibersihkan.

nama_pelanggan	alamat	kota	telp	email
YANTY MAYASARI	GG. TARUN		085659569390	bintang_cianjur@ymail.com
Angela	Jalan Sidom		-	belanglicious@gmail.com
Thian Galuh Intan	Komplek P		087829874149	thian.galuh@rspuriindah.co
Lydia	Jl. Gunung l		0816706919. bb.293c0	bookingan bu ayu
yenny bunny bento	bb		bb	bookingan bu ayu
Murni Juwati Tedja	WARAKU JA		081573202218	murnijuwatitedja@gmail.co
cynthia dewi	permata ser		08161133094	cyndewi@indosat.net.id
jezzyca	Jezzyca Con		bb	bookingan bu ayu
firdila	n		09	forbento@yahoo.com
Puji Dwi Lestari	Jln. Salya nc		08112619712	forbento@yahoo.com
Sandra Tjitra	PT. Kencana		021-5152929	forbento@yahoo.com
farida ariani	jl.s.mahakar		0813 4825 1980	fawryani@yahoo.com
Melissa Tando	Salamander		0818770001	meltando@yahoo.com
Evy Noeriya F	Jln. Masjid /		08569000710	forbento@yahoo.com

Gambar 4.11. Data Pelanggan yang belum dibersihkan

Memilih atribut untuk identifikasi pelanggan yang unik

Diantara pilihan atribut yang ada, yaitu nama_pelanggan, alamat, kota, telepon dan email. Dari pilihan atribut tersebut, dipilih atribut **telepon** sebagai identifikasi pelanggan secara unik. Data pelanggan ini masih didapatkan dari data keseluruhan transaksi yang saat ini berjumlah 10.796 baris data.

Membuang data yang tidak valid

Dari data yang didapatkan, masih terdapat kelemahan dalam dataset pelanggan yaitu banyaknya baris yang tidak valid. Baris ini tidak mengandung format nomor telepon yang berbeda beda sehingga mampu menghambat proses berikutnya. Contoh data yang tidak vaid dan masih terdapat dalam data adalah sebagai berikut

nama_pelanggan	alamat	kota	telp	email
Yunita P.	Komplek Tu		08561352321	puspasariyunita@gmail.com
anita hutty	sby		sudah via FB	sudah via FB
sari cilacap	cilacap		sudah via SMS	sudah via SMS
diah yulianti	jakarta (pes:		sudah via FB	sudah via FB
diah yulianti	jakarta (pes:		sudah via FB	sudah via FB
RENI	PT. COSMO		08151822515	CACHA18@YAHOO.CO.ID
natalia dany parastika	bendosari		sudah via SMS	sudah via SMS
milla (via BB 274f8f9e)	milla (via BE		milla (via BB 274f8f	milla (via BB 274f8f9e)
milla (via BB 274f8f9e)	milla (via BE		milla (via BB 274f8f	milla (via BB 274f8f9e)
mella	jakarta		sudah via SMS	sudah via SMS
milla	jakarta		sudah via SMS	sudah via SMS
cindy	jakarta		087877306090	sudah via SMS
erlinda	jakarta		via BB (2a0023e3)	via BB (2a0023e3)
mila	jakarta		sudah via SMS	sudah via SMS
mila	jakarta		sudah via SMS	sudah via SMS
mila	jakarta		sudah via SMS	sudah via SMS
milla	jakarta		sudah via SMS	sudah via SMS

Gambar 4.12. Contoh data yang tidak valid

Setelah dilakukan seleksi terhadap baris data yang valid, kini jumlah data pelanggan menjadi 5720 baris data dari seluruh transaksi

Menghilangkan meta character pada kolom telepon

Saat ini, kolom telepon masih mengandung berbagai meta-karakter hingga menyebabkan data yang sama masih dibaca sebagai data berbeda, seperti penggunaan telepon dengan (62) ataupun dengan 0, serta pemisahan telepon dengan tanda baca -, :, / dan .. Oleh karena itu seluruh meta karakter yang ada di seluruh data telepon dihilangkan dengan menjalankan perintah query berikut:

```

1. [1] update customer
2.   set telp = replace (telp, '-', '');
3. [2] update customer
4.   set telp = replace (telp, ':', '');
5. [3] update customer
6.   set telp = replace (telp, '/', '');
7. [4] update customer
8.   set telp = replace (telp, '..', '');
9. dst..

```

Kode 4.2. Qyery untuk menghilangkan simbol dan karakter

Setelah query dijalankan maka seluruh meta karakter yang ada di kolom telepon sudah hilang dan memiliki isi atribut yang seragam. Total data yang didapatkan dalam tahap ini masih tetap seperti tahap sebelumnya yaitu 5720 data pelanggan dari keseluruhan transaksi. Data yang saat ini didapatkan berupa nomor telepon dan nomor bb yang bercampur menjadi satu.

Menghilangkan duplikasi data

Setelah no telepon pelanggan sudah diubah dalam format yang sama, kita dapat mengidentifikasi pelanggan yang berbeda dan memberikan nomor identifikasi baru untuk pelanggan tersebut. Hasil yang didapatkan dari query tersebut yaitu berupa data customer dengan jumlah 963 pelanggan yang unik

```
1. ALTER IGNORE TABLE jobs
2. ADD UNIQUE INDEX idx_cust (nama_pelanggan, alamat, telepon, email);
```

Kode 4.3. Query untuk menghilangkan duplikasi data

4.1.3.2 Pra-proses data order

Sedangkan untuk kebutuhan data penjualan yang berisi data pelanggan dan produk yang dibeli masing-masing, perlu dilakukan seleksi data dengan menggabungkan dua buah tabel dari database yaitu pada tabel pengiriman (v_transaksi) dan tabel transaksi (v_detail_transaksi). Dengan demikian dapat diketahui siapa pelanggan yang melakukan transaksi tersebut.

```
1. SELECT DISTINCT a.id_transaksi, nama_pelanggan,
   telpon, id_barang, quantity FROM v_transaksi a RIGHT
   JOIN transaksi_detail b ON a.id_transaksi= b.
   id_transaksi
```

Kode 4.4. Query penggabungan data pelanggan dan order

Dari query pada Kode 4.4. Query penggabungan data pelanggan dan order menghasilkan 23.634 baris data yang menunjukkan transaksi dengan identitas pelanggan dan produk yang dibelinya. Hasil eksekusi query dapat dilihat pada Gambar 4.13

id_transaksi	nama_pelanggan	telp	id_barang	quantity
8574	Bu Yuanita Anggia Sari	0878-8839-9918	817	1
8575	Bu Arumma	082112611441	1037	1
8576	Bu Niken	0812-3588-979 atau 0	1154	1
8576	Bu Niken	0812-3588-979 atau 0	1100	1
8577	Bu Anggi	0818-7488-477	586	1
8577	Bu Anggi	0818-7488-477	857	50
8578	Bu Renny Setiana	0853-4942-8877	1059	1

Gambar 4.13. Hasil penggabungan data

Hasil order ini dimasukkan ke dalam database tersendiri untuk memudahkan dalam pengolahan data selanjutnya.

wp_listorder	
id:	int
id_transaksi:	int
username:	varchar(27)
product_name:	varchar(57)
price:	int
quantity:	int
status:	varchar(5)

Gambar 4.14. Database gabungan pelanggan dan order

Setelah data order berhasil di dapatkan, maka dilakukan agregasi data untuk mengetahui *concurrency order* dari setiap pelanggan terhadap suatu produk. Berikut ini merupakan query yang dieksekusi untuk menghasilkan agregasi dari nilai penjualan berulang.

Hasil dari eksekusi query ditampilkan dalam Gambar 4.15:

```
1. SELECT id_transaksi, username, product_name, count (*) as recurrence_order
```

```
2. from wp_listorder GROUP BY username, product_name
```

Kode 4.5. mencari recurrence order

id_transaksi	username	product_name	recurrence_order
1105	EliaAzaliaRental	Rice Mold - Set Animals Hello Kitty Small	4
7302	Eliece	Catering Box Hiro For Sushi ST-2	1
3999	ELIS	Bento Book - ibento edisi 2	3
1432	Elis	Bento Cutter - Multi Bear Rabbit	4
3999	ELIS	Bento Cutter - Vege Paku Melody	3
1432	Elis	Egg Shaper - Pancake Ring Rabbit	4
3999	ELIS	Nori Kemasan Ulang Takaoka	3
1432	Elis	Nori Punch - KMC Smile Emoticon Shape	4
3999	ELIS	Nori Punch - Set Doraemon with Puncher	3
3999	ELIS	Sauce Mini Cup Panda & Rabbit	3

Gambar 4.15. Data pembelian beserta recurrence order

4.2 Perancangan aplikasi klien

4.2.1 Analisis kebutuhan aplikasi klien

Kebutuhan aplikasi klien didapat berdasarkan proses penjualan yang dilakukan oleh toko online forbento sebagai objek studi kasus dalam penelitian ini. Saat ini forbento memiliki website yang menggunakan platform wordpress dan woocommerce sebagai media penjualannya (www.forbento.com). Sistem rekomendasi yang ada dalam website forbento masih dibuat berdasarkan tema wordpress yang digunakan. Dalam hal ini rekomendasi masih berupa rekomendasi produk dalam satu kategori. Dengan proses bisnis yang sederhana, pendaftaran pelanggan tidak dilakukan oleh masing-masing pelanggan namun dilakukan oleh admin. Oleh karena itu, dalam penelitian ini, lingkungan klien dibuat dengan model dan proses yang sama dengan toko online Forbento. Klien ini digunakan untuk mengumpulkan dan menampilkan semua data dalam bentuk aplikasi dan mensimulasikan jalannya proses rekomendasi dari pengguna

hingga mendapatkan hasil rekomendasi dari server yang akan dibuat.

Untuk melakukan perancangan aplikasi klien dengan baik, dilakukan analisis kebutuhan hingga didapatkan daftar kebutuhan berikut untuk aplikasi klien:

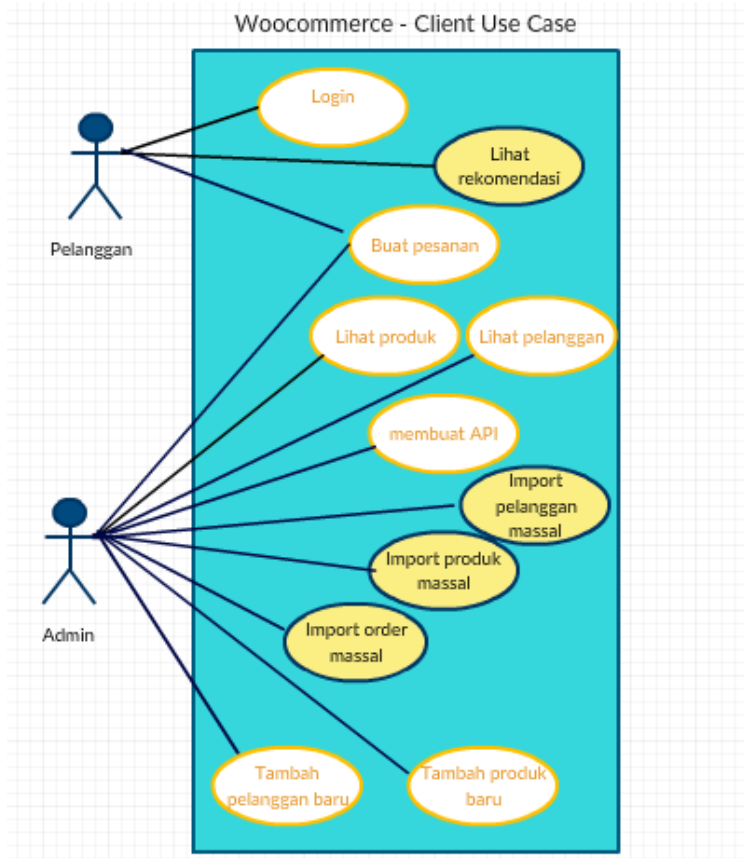
Kebutuhan fungsional aplikasi klien:

- Aplikasi dapat menangani import produk secara massal
- Aplikasi dapat menangani import pelanggan secara massal
- Aplikasi dapat menangani import order otomatis secara massal
- Aplikasi dapat menampilkan produk
- Aplikasi dapat menangani proses login dan logout pelanggan
- Aplikasi dapat menampilkan hasil rekomendasi berdasarkan informasi pelanggan yang sedang login
- Admin dapat menambah pelanggan baru/member baru
- Admin dapat menambah produk baru
- Admin dapat menambahkan pelanggan baru
- Admin dapat menambahkan pesanan baru

4.2.2 Desain Sistem

Desain sistem pada use case diagram dibawah ini mencakup beberapa fungsi/fitur utama yang tersedia dalam lingkungan wordpress dan woocommerce yaitu pada use case berwarna putih.

Sedangkan fungsi tambahan yang perlu dibuat, ditampilkan dalam point use case berwarna kuning yaitu import pelanggan, produk dan order secara massal, serta use case melihat rekomendasi.



Gambar 4.16. Desain use case client

4.2.3 Perancangan database klien

Database klien merupakan database yang di generate dari wordpress dengan beberapa tambahan database untuk menyimpan data hasil praproses. Daftar database yang dihasilkan dari aplikasi klien yaitu terdapat 37 tabel yang menampung seluruh informasi produk, pelanggan, order serta seluruh meta data dari wordpress dan woocommerce.

Berikut ini merupakan tabel deskripsi dari wordpress yang digunakan untuk mengumpulkan data yang akan diolah menjadi API.

Tabel 4.1. Data dictionary wordpress

Nama Tabel	Deskripsi	Navigasi Area pada interface wordpress
wp_commentmeta	Setiap komentar menampilkan informasi yang disebut meta data dan disimpan di wp_commentmeta.	Administratio n > Comments > Comments
wp_comments	Komentar di dalam WordPress disimpan di tabel wp_comments.	Administratio n > Comments > Comments
wp_links	Wp_links menyimpan informasi yang terkait dengan link yang masuk ke dalam fitur Links dari WordPress.	Administratio n > Links > Add New Administratio n > Links > Links
wp_options	Pilihan yang diatur di bawah panel Administrasi> Pengaturan disimpan di tabel wp_options. Lihat Option Reference untuk option_name dan nilai default.	Administratio n > Settings > General

wp_postmeta	Setiap posting menampilkan informasi yang disebut meta data dan disimpan di wp_postmeta. Beberapa plugin mungkin menambahkan informasi mereka sendiri ke tabel ini.	Administratio n > Posts > Add New
wp_posts	Inti dari data WordPress adalah post. Ini disimpan di tabel wp_posts. Juga Halaman dan item menu navigasi disimpan dalam tabel ini.	Administratio n > Appearance > Menus
wp_terms	Kategori untuk kedua posting dan link dan tag untuk posting ditemukan di dalam tabel wp_terms.	Administratio n > Posts > Post Tags
wp_term_relationships	Pos dikaitkan dengan kategori dan tag dari tabel wp_terms dan asosiasi ini dipertahankan di tabel wp_term_relationships. Hubungan link ke kategori masing-masing juga disimpan dalam tabel ini.	Administratio n > Posts > Post Tags

wp_term_taxonomy	Tabel ini menjelaskan taksonomi (kategori, tautan, atau tag) untuk entri dalam tabel wp_terms.	Administratio n > Posts > Post Tags
wp_usermeta	Setiap pengguna menampilkan informasi yang disebut meta data dan disimpan di wp_usermeta	Administratio n > Users
wp_users	Daftar pengguna dikelola dalam tabel wp_users.	Administratio n > Users

Sedangkan untuk plugin woocommerce yang diinstall dalam lingkungan wordpress akan menghasilkan tabel dengan deskripsi sebagai berikut:

Tabel 4.2. Data dictionary woocommerce

Nama Tabel	Deskripsi
Woocommerce_sessions	Menyimpan data sesi pelanggan, seperti keranjang belanja.
Woocommerce_api_keys	Menyimpan API key yang digunakan untuk REST API.
woocommerce_attribute_taxonomies	Menyimpan nama taksonomi atribut global untuk produk.
woocommerce_downloadable_product_permissions	Menyimpan izin akses produk untuk produk yang dapat didownload (diberikan setelah pembelian).
woocommerce_order_items	Menyimpan item baris yang terkait dengan pesanan.
woocommerce_order_	Menyimpan data meta tentang item baris pesanan.

itemmeta	
woocommerce_ tax_ rates	Menyimpan tarif pajak yang Anda tentukan di area admin.
woocommerce_ tax_rate_ locations	Menyimpan lokasi (kode pos dan kota) yang terkait dengan tarif pajak di atas.
woocommerce_ shipping_zones	Menyimpan zona pengiriman yang Anda buat di area pengaturan.
woocommerce_ shipping_zone_ locations	Menyimpan lokasi yang terkait dengan zona pengiriman Anda.
woocommerce_ shipping_ zone_ methods	Menyimpan metode pengiriman yang terkait dengan zona pengiriman Anda.
woocommerce_ payment_ tokens	Menyimpan token pembayaran pelanggan (digunakan oleh gateways).
woocommerce_ payment_ tokenmeta	Menyimpan meta data tentang token pembayaran.

4.2.4 Perancangan pengujian klien

Pengujian aplikasi klien dilakukan dengan melakukan unit test pada setiap fungsi yang berhasil dieksekusi dengan mengecek kesesuaian output dengan hasil yang diharapkan . Pengujian dilakukan dengan metode black box dan mencatat error yang terjadi di setiap iterasi.

4.3 Perancangan aplikasi server

Aplikasi server dibuat sebagai sistem utama yang memproses rekomendasi serta berperan sebagai penghubung antara lingkungan client dan database neo4j.

Secara lebih mendetail, aplikasi server bertanggung jawab untuk mengambil data secara berkala dari aplikasi client yaitu data produk, data pelanggan dan data order dalam bentuk json. Aplikasi server melakukan update data untuk memastikan bahwa data yang ada di dalam database server telah diperbaharui. Dengan demikian, web service akan dapat mengambil hasil rekomendasi dari data yang terbaru.

4.3.1 Analisis kebutuhan aplikasi server

Dalam penelitian ini, terdapat beberapa kebutuhan fungsional yang perlu dibuat dalam aplikasi server.

Kebutuhan fungsional aplikasi server:

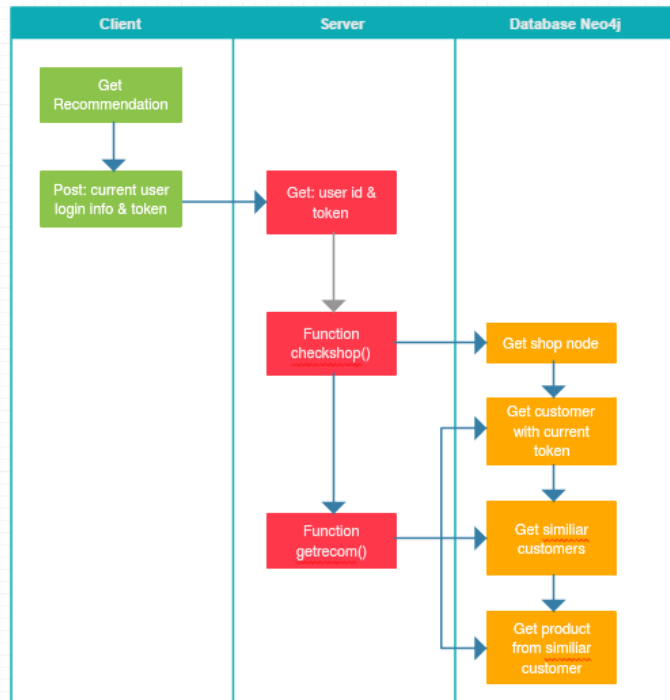
- Aplikasi dapat menangani proses registrasi member
- Aplikasi dapat menangani proses pendaftaran toko
- Aplikasi dapat menangani proses login member yang telah mendaftar
- Aplikasi dapat melakukan update data selama 3 jam sekali
- Aplikasi dapat membuat API untuk mengambil data produk
- Aplikasi dapat membuat API untuk mengambil data order
- Aplikasi dapat membuat API untuk mengambil data pelanggan
- Aplikasi dapat mengolah data menjadi hasil rekomendasi kepada aplikasi klien
- Member dapat melihat daftar produk terakhir
- Member dapat melihat toko yang sudah didaftarkan
- Member dapat melihat pelanggan yang terdaftar
- Member dapat mengecek token
- Member dapat mengecek profil

4.3.2 Desain sistem server

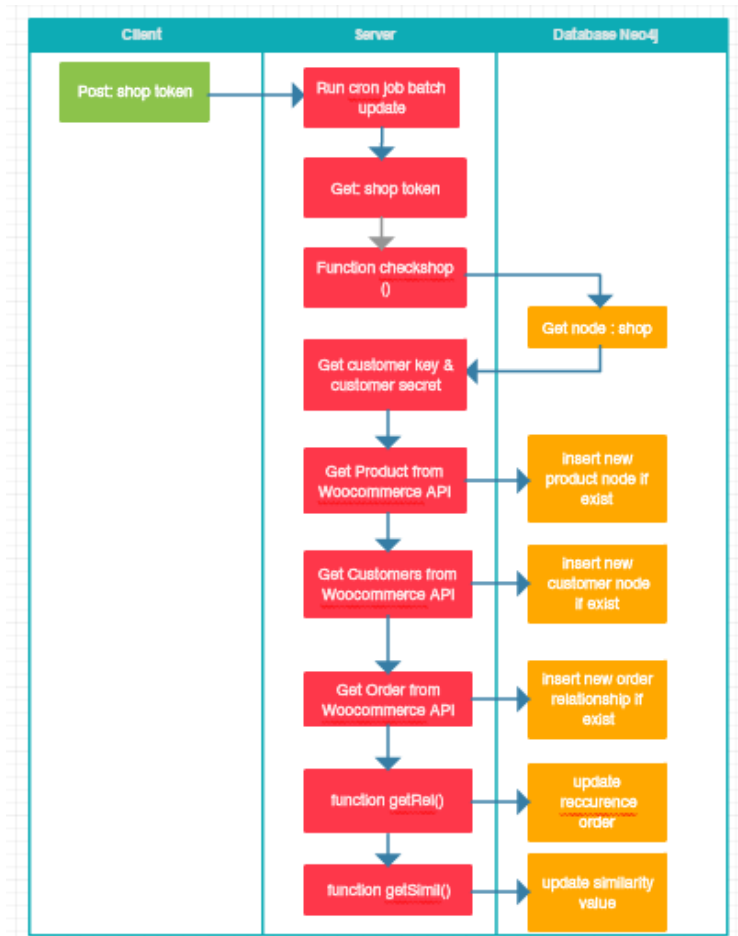
Desain sistem pada use case diagram dibawah ini mencakup beberapa fungsi/fitur utama yang akan dirancang pada sistem server yang terhubung dengan database

4.3.2.1 Desain Alur Sistem

Desain alur sistem pada web service dibagi menjadi 2 proses utama, yaitu proses pengambilan rekomendasi yang digambarkan dalam Gambar 4.17 dan proses update data pada Gambar 4.18



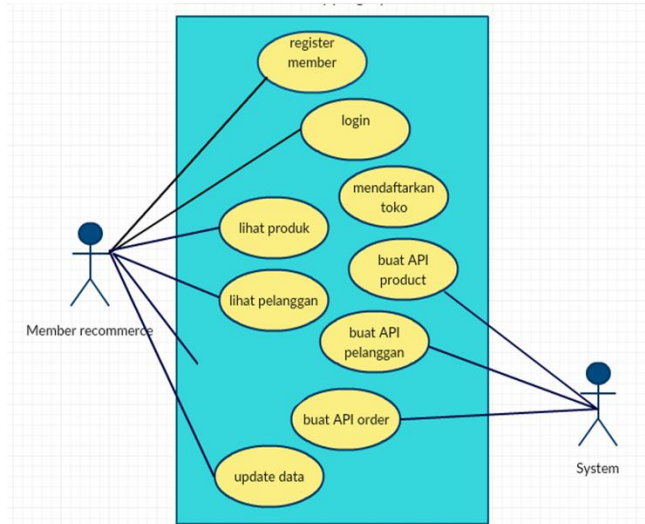
Gambar 4.17. Swimlane generate rekomendasi pada server



Gambar 4.18. Swimlane Update data server

4.3.2.2 Desain Use Case

Sedangkan berikut ini dijabarkan desain use case yang dapat dilakukan oleh member web service terhadap sistem pada Gambar 4.19



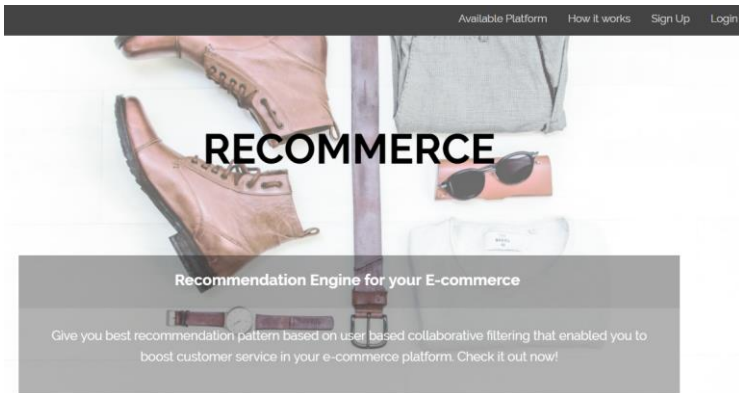
Gambar 4.19. Desain use case server

4.3.2.3 Desain Interface

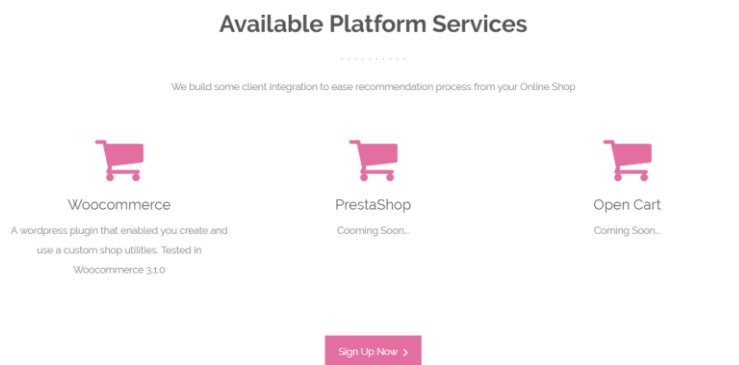
Dari seluruh proses tersebut akan dihasilkan beberapa user interface berikut:

a. Halaman Home Recommerce

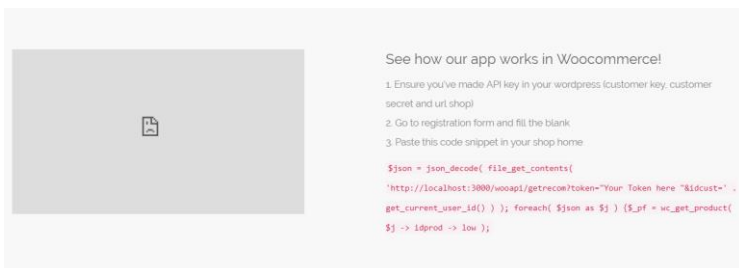
Halaman home recommerce merupakan halaman pengenalan terkait web service recommerce beserta layanan yang diberikannya. Tampilan landing page dapat dilihat pada Gambar 4.20, Gambar 4.21, Gambar 4.22.



Gambar 4.20. Halaman Home Recommerce section top_content



Gambar 4.21. Halaman home recommerce section feature



Gambar 4.22. Halaman home recommerce section how to work

Available Platform API How it works FAQ

Recommerce

Recommendation Engine for your E-commerce

Give you best recommendation pattern based on user based collaborative filtering that enabled you to boost customer service in your e-commerce platform.

Sign Up Now / Login

Fill in the form below to get recommerce access.

username

Email

password

Register!

Gambar 4.23. Halaman Register Member

Available Platform API How it works FAQ

Recommerce

Recommendation Engine for your E-commerce

Give you best recommendation pattern based on user based collaborative filtering that enabled you to boost customer service in your e-commerce platform.

Login Now / Sign up

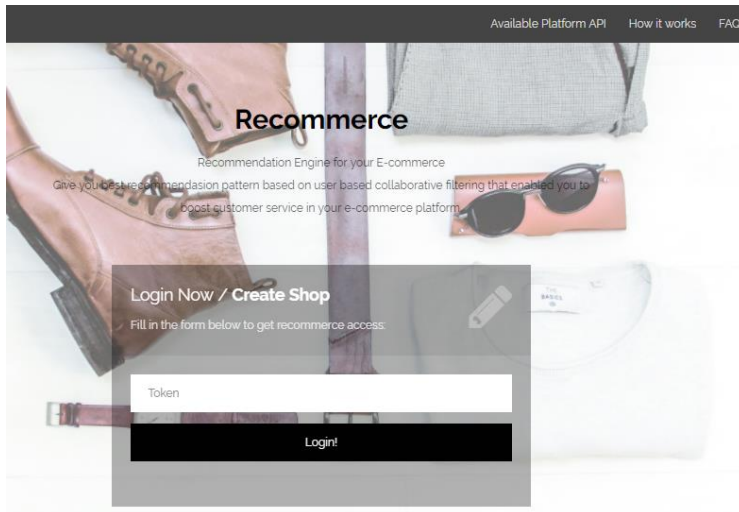
Fill in the form below to get recommerce access.

Email

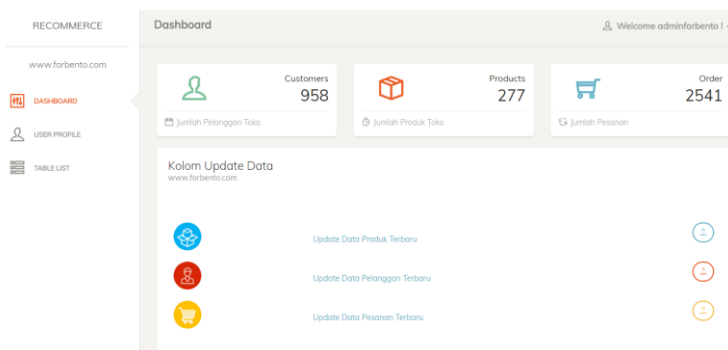
password

Login!

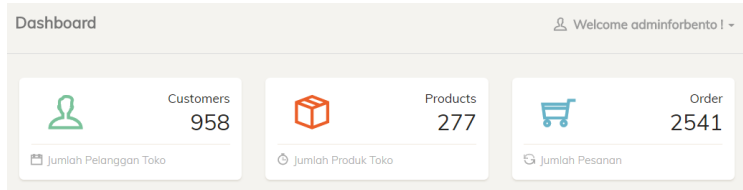
Gambar 4.24. Halaman Login Member



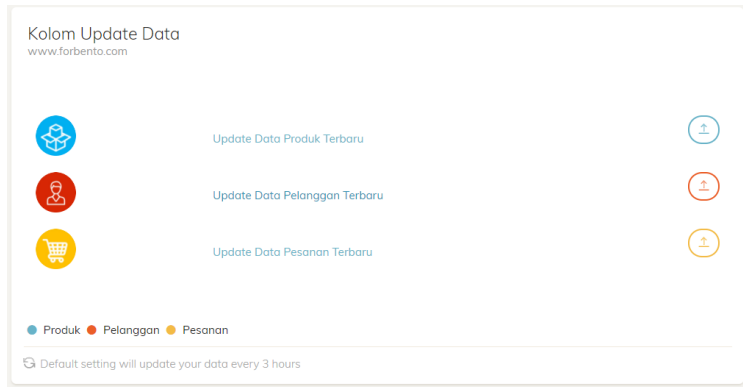
Gambar 4.25. Halaman Login Toko



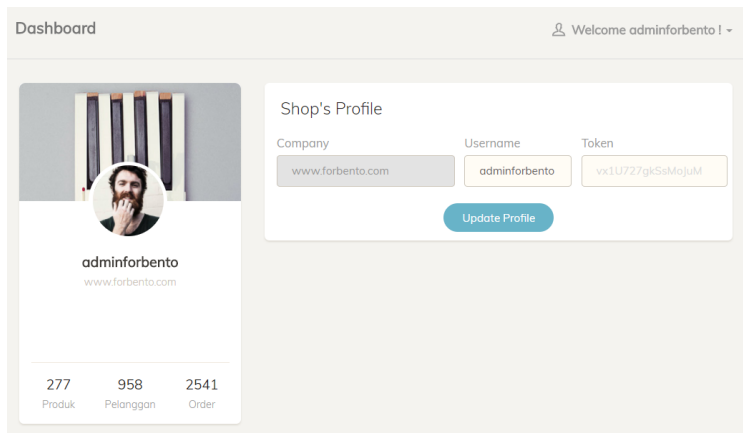
Gambar 4.26. Halaman Dashboard Awal



Gambar 4.27. Halaman Dashboard section product-information



Gambar 4.28. Halaman Dashboard section update data



Gambar 4.29. Halaman Profil Toko Member

RECOMMERCE

www.forbento.com

DASHBOARD

USER PROFILE

TABLE LIST

Table on Plain Background
Here is a subtitle for this table

ID	Name	Salary	Country	City
1	Dakota Rice	\$36,738	Niger	Oud-Turnhout
2	Minerva Hooper	\$23,789	Curaçao	Sinaai-Waas
3	Sage Rodriguez	\$56,142	Netherlands	Baileux
4	Philip Chaney	\$38,735	Korea, South	Overland Park
5	Doris Greene	\$63,542	Malawi	Feldkirchen in Kärnten
6	Mason Porter	\$78,615	Chile	Gloucester

Gambar 4.30. Halaman Daftar Produk

4.3.3 Perancangan database server

Database server dirancang menggunakan basis data no sql berbasis graf neo4j untuk memproses data yang didapatkan dari aplikasi client sehingga dapat menghasilkan rekomendasi yang sesuai

Permodelan data yang digunakan dalam penelitian ini merujuk pada model 2 layer database graf yang dirujuk dari penelitian Zan Chung Huan yang merepresentasikan model antara produk, pelanggan dan transaksi yang terjadi diantara keduanya

Seluruh proses bisnis yang akan dilakukan di dalam server akan dijadikan acuan dalam merancang database dalam server. Gambar 4.31 di bawah ini merupakan skema database graf yang akan dibuat dalam aplikasi server. Secara keseluruhan, objek node dalam web service berjumlah 5 jenis node. Masing masing node merepresentasikan jenis node nya masing masing sehingga dalam hal ini node sejenis dapat mencapai jumlah 35 miliar.

Berikut ini merupakan daftar node/ object dalam server

Tabel 4.3. Daftar Node dalam server

Jenis node	Deskripsi
Member	Setiap admin toko yang mendaftarkan diri melalui website recommerce untuk

	mendapatkan akses ke web server
Shop	Toko yang dimiliki oleh setiap member. Bisa berjumlah lebih dari 1 toko
Products	Setiap produk yang ada di dalam sebuah toko.
Customers	Pelanggan yang dimiliki oleh member dalam sebuah toko

Sedangkan penjelasan atribut masing – masing node akan dijabarkan dalam tabel berikut

Tabel 4.4. Node member dalam graph

Node Member	
Nama Atribut	Deskripsi
Id	Nomor identifikasi node member
Membername	Username member yang telah didaftarkan dalam sistem recommerce
Email	Email yang telah didaftarkan dalam sistem recommerce
Password	Password sebagai autentikasi login user

Tabel 4.5. Node Shop dalam graph

Node Shop	
Nama Atribut	Deskripsi
Id	Nomor identifikasi node shop
Urlshop	Alamat url toko yang dapat diakses
Type	Tipe platform e commerce yang digunakan, seperti woocommerce, prestashop dll Saat ini masih tersedia tipe 1 yaitu woocommerce
Customer key	Sebagai akses web service untuk mengambil data dari aplikasi klien

Customer secret	Sebagai akses web service untuk mengambil data dari aplikasi klien
Totalprod	Jumlah produk terakhir yang berada di dalam database
Totalcust	Jumlah pelanggan terakhir yang berada di dalam database
Totalord	Jumlah pesanan terakhir yang berada di dalam database

Tabel 4.6. Node customer dalam graph

Node Customers	
Nama Atribut	Deskripsi
Custid	Nomor identifikasi node customer
Name	Nama customer
Token	Identifikasi pelanggan dari toko

Tabel 4.7. Node produk dalam graph

Node Products	
Nama Atribut	Deskripsi
Custid	Nomor identifikasi node produk
Name	Nama produk dari sebuah toko
Price	Harga produk dalam sebuah toko
Token	Identifikasi produk dari toko

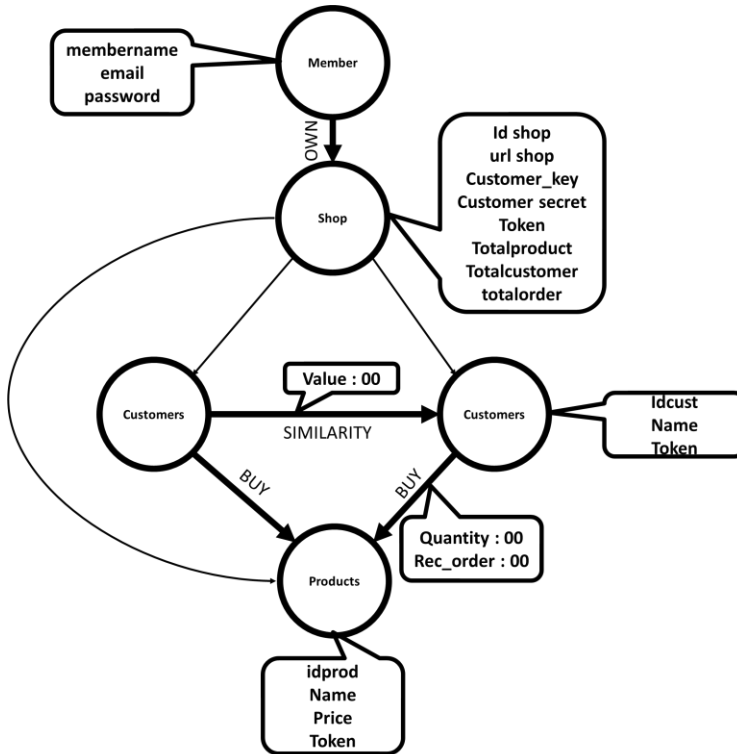
Selain itu hubungan antar node memiliki makna yang juga merepresentasikan proses bisnis dari web service digambarkan berupa edge antara node satu dengan node yang lain

Tabel 4.8. Hubungan antar graph

Nama edge	Repreentasi hubungan	Deskripsi
Has	Member-[:HAS]->Shop	Kepemilikan toko terhadap masing masing member
Buy	Customers-[:BUY]->Products	Mengidentifikasi pembelian yang

		<p>dilakukan oleh setiap pelanggan Relationship :BUY memiliki beberapa atribut diantaranya :</p> <p>-quantity: jumlah barang yang dibeli dalam satu transaksi</p> <p>-ordervalue: jumlah recurrence order yang telah dilakukan oleh pelanggan terhadap barang tersebut.</p>
Similarity	Customers – [:SIMILARITY}->Customers	Nilai kemiripan antar pelanggan

Dari seluruh penjabaran struktur data yang ada di dalam server, skema database server digambarkan dalam Gambar 4.31 di bawah ini



Gambar 4.31. Desain Model Graph

4.3.4 Perancangan library pada server

Dalam implementasinya aplikasi server menggunakan node js untuk pengembangannya, oleh karena itu dilakukan perancangan terhadap library yang akan diimplementasikan dalam server yang akan dibuat:

Tabel 4.9. Daftar library node js yang akan digunakan

Nama Library	Deskripsi
Path	Modul path menyediakan utilitas untuk bekerja dengan file path dan direktori

express	Kerangka kerja web minimalis yang cepat dan tidak terbuka, untuk node.
Morgan	HTTP request logger middleware untuk node.js
Neo4j	database driver untuk Neo4j versi 3.0.0+.
Woocommerce	Wrapper Node.js untuk REST API WooCommerce. Mudah berinteraksi dengan REST REST WooCommerce menggunakan library ini.
Body-parser	Melakukan parsing dari request bodies yang ada dalam sebuah middleware sebelum handlers, terdapat pada bagian setelah req.body property.
Json-file	Untuk membaca atau membuat file json

4.3.5 Perancangan pengujian web service

Pengujian web service dilakukan dengan melakukan unit test pada setiap fungsi yang berhasil dieksekusi dengan mengecek kesesuaian output dengan hasil yang diharapkan . Pengujian dilakukan dengan metode black box dan mencatat error yang terjadi di setiap iterasi.

4.4 Perancangan rekomendasi

4.4.1 Perancangan algoritma rekomendasi

Untuk menghasilkan rekomendasi digunakan pendekatan algoritma customer based collaborative filtering yang

menghitung nilai kedekatan antar user dari catatan pembelian yang telah dilakukannya dan catatan recurrence order yang pernah terjadi pada setiap pembeliannya. Seluruh pemrosesan ini dilakukan dalam query cypher yang langsung di eksekusi dalam database neo4j. Sedangkan untuk menghitung nilai kemiripan antar user digunakan 2 buah metode untuk perbandingan yaitu cosine similarity dan euclidian distance. Kedua metode ini dipilih berdasarkan Penelitian Movie Lens oleh Michael Ekstrand yang menyatakan metode Cosine Similarity jauh lebih baik daripada Pearson setelah dilakukan pengkajian ulang dengan mempertimbangkan vektor yang berpusat pada rata-rata. Sedangkan di sisi lain penelitian yang dilakukan oleh A. Jeyasekar membuktikan bahwa penggunaan metode euclidian distace dalam mengukur kemiripan user jauh lebih efektif daripada KL Divergences. Oleh karena itu, dalam penelitian ini digunakan pendekatan dari metode terbaik sebagai perbandingannya.

4.4.2 Perancangan pengujian rekomendasi

Pengujian rekomendasi dilakukan dengan melakukan validasi terhadap expert toko online forbento untuk melakukan simulasi rekomendasi pada 4 skenario yang berbeda. Selain itu juga dilakukan pengujian ketepatan rekomendasi pada customer forbento berdasarkan riwayat pembelian yang dilakukannya. Tujuan Pengujian rekomendasi dilakukan untuk mengetahui hal berikut:

1. Mengetahui ketepatan rekomendasi yang dihasilkan dari algoritma collaborative filtering terhadap pelanggan Forbento
2. Membandingkan penggunaan perhitungan kesamaan pelanggan yang berbeda antara cosine similarity dan euclidian distance untuk menghasilkan rekomendasi yang lebih baik
3. Mengetahui faktor faktor yang mempengaruhi ketepatan rekomendasi dalam studi kasus Forbento

BAB V

IMPLEMENTASI

Pada bab ini, akan dijelaskan mengenai implementasi dari perancangan yang telah dilakukan sesuai dengan metode pengembangan yang dibuat. Bagian implementasi akan menjelaskan mengenai lingkungan implementasi, pembuatan fitur-fitur aplikasi dalam bentuk kode, serta pengujian aplikasi.

5.1 Lingkungan Implementasi

Pengembangan aplikasi ini menggunakan komputer dengan spesifikasi pada Tabel 5.1 berikut:

Tabel 5.1. Spesifikasi lingkungan implementasi

Prosesor	Intel® Core™ i5-7200U CPU @ 2.50GHz
Memory	4 GB RAM
Sistem Operasi	Windows 10 Pro
System type	64-bit Operating System, x64-based processor

Aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti editor, database, server; bahasa pemrograman, dan library yang disajikan dalam Tabel 5.2 berikut:

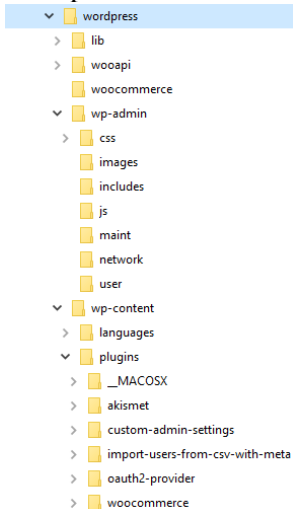
Tabel 5.2. Lingkungan teknologi yang digunakan

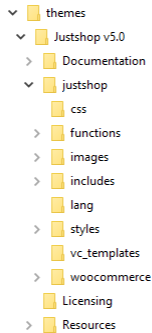
Webserver	Apache 2.4, Node JS
Bahasa Pemrograman	PHP 7.1, Javascript, Cypher
Database	<ul style="list-style-type: none">• MySQL (client)• Neo4j (web service)

Editor (IDE)	Atom , Sublime Text
Browser	Google Chrome 56
Library	<ul style="list-style-type: none"> • Node js • Express Template for Node js • Wordpress • Woocommerce • Klooon php woocommerce

5.2 Pembangunan Client

Berikut ini dijabarkan file direktori dari pembangunan aplikasi klien pada Gambar 5.1

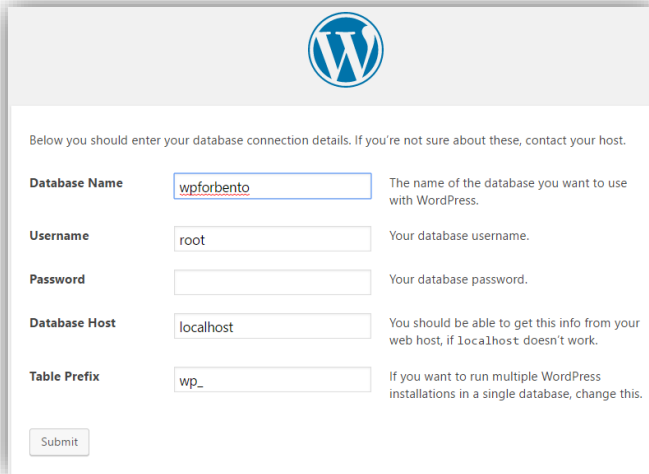




Gambar 5.1. Direktori Client

5.2.1 Instalasi Wordpress

Untuk menyesuaikan lingkungan implementasi Forbento maka dilakukan instalasi wordpress terlebih dahulu, adapun paket wordpress dapat diunduh secara langsung pada situs resminya www.wordpress.org. Selanjutnya file dapat diakses melalui url localhost tempat file wordpress diletakkan. Setelah melewati konfigurasi umum wordpress, dibuat sebuah database yang digunakan untuk seluruh konfigurasi data wordpress yang akan diinstall.



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="wpforbento"/>	The name of the database you want to use with WordPress.
Username	<input type="text" value="root"/>	Your database username.
Password	<input type="password"/>	Your database password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost doesn't work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.

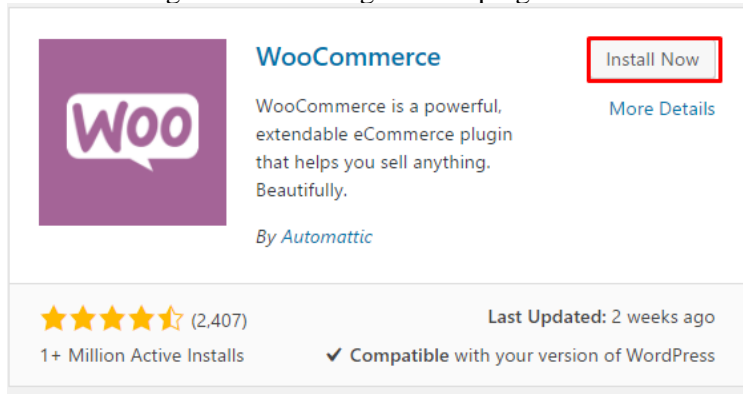
Gambar 5.2. Halaman Konfigurasi awal wordpress

Lalu seluruh informasi database dimasukkan dalam deskripsi koneksi database pada setup wordpress seperti nama database, username, password, database host, dan prefix tabel yang akan digunakan seperti Gambar 5.2 di atas. Langkah selanjutnya yaitu autentikasi user dengan memasukkan judul site, username, password dan email administrator. Setelah semua informasi yang dibutuhkan telah dimasukkan, instalasi wordpress akan berhasil.

5.2.2 Instalasi Woocommerce

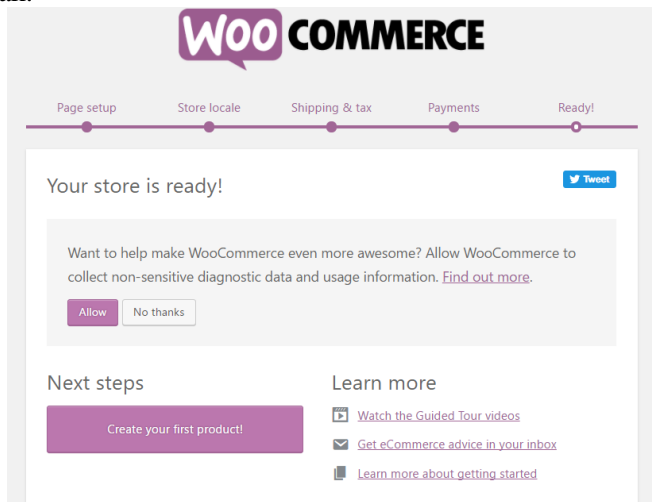
Instalasi woocommerce dibutuhkan untuk mengaktifkan fitur e-commerce pada situs wordpress. Dengan adanya fitur ini, baik penjual maupun pembeli dapat melakukan transaksinya secara langsung melalui website wordpress yang telah dibuat sebelumnya tanpa harus membuat aplikasi baru untuk penjualan. Untuk melakukan instalasi woocommerce, masuk ke dalam menu administrator, pada sidebar, klik menu plugin dan

tambahkan woocommerce sebagai plugin baru. Sistem akan otomatis menginstall dan mengaktifkan plugin woocommerce.



Gambar 5.3. Install plugin woocommerce

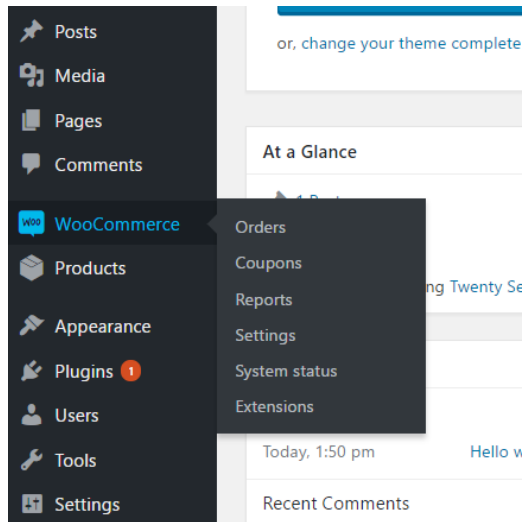
Proses instalasi menggunakan user interface yang mudah sehingga dapat dimengerti dengan baik. Konfigurasi yang diperlukan untuk menginstall woocommerce seperti konfigurasi umum toko, satuan pembayaran, pengiriman dan pajak.



Gambar 5.4. tampilan sukses install woocommerce

Gambar .. merupakan tampilan halaman administrator setelah plugin woocommerce diaktifkan. Pada menu woocommerce,

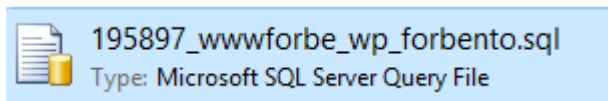
dapat dilihat daftar order, kupon, pengaturan toko, serta informasi product dan pelanggan.



Gambar 5.5. Tampilan menu bar setelah install woocommerce

5.2.3 Migrasi Data

Setelah klien berhasil dibuat maka perlu dilakukan migrasi data pelanggan dan migrasi data order mengingat data produk sudah tersimpan secara otomatis dalam data wordpress yang didapatkan. Karena database klien masih belum memuat data transaksi apapun, dilakukan import file sql di bawah ini dari panel phpmyadmin.



Gambar 5.6. File sql wodpress forbento

File to import:

Dapat berupa berkas terkompresi (gzip, bzip2, zip) atau tidak.
 Nama berkas terkompresi harus diakhiri dengan **[format].[kompresi]**. Contoh: **.sql.zip**

Telusuri komputer Anda: 195897_wwwf...rbento.sql (Batas ukuran: 1,280MB)

You may also drag and drop a file on any page.

Set karakter berkas:

Gambar 5.7. Import file sql ke database wordpress

Setelah file diimport maka kita akan menemukan error karena adanya crash konfigurasi awal yang dibuat dengan konfigurasi di dalam file sql yang telah ada. Untuk mengatasinya, kita perlu memastikan bahwa semua tabel dalam database telah kosong yaitu dengan menggunakan fungsi truncate table.

1. **TRUNCATE TABLE** wp_comments
2. **TRUNCATE TABLE** wp_commentmeta
3. **TRUNCATE TABLE** wp_user
4. **TRUNCATE TABLE** wp_usermeta

Kode 5.1. Query untuk mengosongkan tabel

5.2.4 Pembuatan Plugin Import Customer

Untuk memindahkan data pelanggan ke aplikasi client yang baru, maka perlu dilakukan instalasi plugin yang tersedia di wordpress yaitu plugin import users from CSV with meta



The screenshot shows the WordPress plugin directory page for the 'Import users from CSV with meta' plugin. The plugin icon is a purple square with a white document icon and the text 'Import users from CSV with meta'. The title 'Import users from CSV with meta' is displayed in blue. Below the title, there is a description: 'A plugin to import users using CSV files to WP database automatically including custom user meta'. The author is listed as 'Oleh codecton'. The plugin has a rating of 4.5 stars (87 reviews) and over 20,000 active installations. It was last updated 1 week ago and is compatible with the current version of WordPress.

Gambar 5.8. Install plugin import user from CSV with meta

Untuk melakukan import, format data harus diunggah dengan format csv dan sesuai format yang telah ditentukan yaitu username, email, password, address, city, state, zipcode, country, user_url. Tampilan data yang telah diformat digambarkan dalam tampilan gambar di bawah ini

Username	Email	Password	Address	City	State	Zipcode	country	user_url
YunifarFerariani	default@default.com	4	admin	Jl. Mulyosari BPB No 13	Surabaya	Jawa Tim		
LalaPutri	default@default.com	5	admin	Jl. Mulyosari BPB No 14	Surabaya	Jawa Timur	60111	
Yovita	default@default.com	8	admin	Jl. Mulyosari BPB No 15	Surabaya	Jawa Timur	60111	
DewiCandra	default@default.com	10	admin	Jl. Mulyosari BPB No 16	Surabaya	Jawa Timur		
BuArieSetyorini	default@default.com	11	admin	Jl. Mulyosari BPB No 17	Surabaya	Jawa Tim		
anny	default@default.com	12	admin	Jl. Mulyosari BPB No 18	Surabaya	Jawa Timur	60111	
HannyWijaya	default@default.com	15	admin	Jl. Mulyosari BPB No 19	Surabaya	Jawa Timu		
BuRaraAyuSetyaningrum(Rara)	default@default.com	16	admin	Jl. Mulyosari BPB No 20	Su			
YoyokMediana	default@default.com	19	admin	Jl. Mulyosari BPB No 21	Surabaya	Jawa Tim		

Gambar 5.9. Contoh data CSV yang telah diolah

Konfigurasi umum import juga dapat disesuaikan seperti pengaturan role dari setiap user yang diimport, dan pilihan update dari data user yang sudah tersimpan sebelumnya seperti Gambar 5.10. Tampilan plugin import CSV

Gambar 5.10. Tampilan plugin import CSV

5.2.5 Pembuatan Plugin Import Order

Untuk memindahkan data order ke aplikasi client, wordpress tidak menyediakan plugin yang spesifik untuk hal tersebut, oleh karena itu dalam hal ini perlu dilakukan pembuatan plugin secara custom.

```

1. <?php
2. /*
3. Plugin Name: Custom Administration Page
4. Plugin URI: http://www.uri.com
5. Description: a custom plugin
6. Version: 1.2
7. Author: Mr. X
8. Author URI: authoruri.com
9. License: GPL2
10. */
11. ?>

```

Kode 5.2. Konfigurasi awal plugin

Untuk membuat plugin di wordpress, dibuat satu folder dan buat sebuah file. Letakkan folder pada folder wp-content/plugins, dan tambahkan file dengan nama pluginnya misalkan customAdministrationPage.php. Buka text editor dan masukkan informasi ini di dalamnya:

Dari seluruh informasi di atas yang wajib diisi adalah nama plugin. Hanya dengan hal tersebut, pada bagian back-end plugin yang telah dibuat dapat diaktifkan.

Pada kode berikut ini dijelaskan bahwa fungsi dibuat pada halaman baru dari sub menu, dengan *new Submenu(new Submenu_Page())*. Pada halaman tersebut, dibuat suatu fungsi plugin yang menambahkan proses *woocommerce_checkout_process* dan *create_vip_order* yang mengambil database dari tabel *wp_listorder* yang belum diimport sejumlah 100 secara berkala. Data dari tabel *wp_listorder* akan diimport ke dalam tabel konfigurasi

wordpress untuk membuat order. Plugin akan mencocokkan username dari tabel wp_listorder dengan data pelanggan yang ada di dalam wordpress serta mencocokkan data nama produk dari wp_listorder dengan data produk di katalog wordpress dan mencatatnya sebagai order baru

5.2.6 Menampilkan rekomendasi

Mekanisme klien untuk menampilkan rekomendasi, dimulai dengan pelanggan yang login ke dalam woocommerce client. Lalu woocommerce akan mengambil informasi pelanggan yang sedang login dan menampilkan rekomendasi khusus untuknya pada section khusus rekomendasi pada woocommerce.

```

1. foreach ( glob( plugin_dir_path( __FILE__ ) . 'a
   dmin/*.php' ) as $file ) {
2. include_once $file;
3. }
4.
5. add_action( 'plugins_loaded', 'tutsplus_custom_a
   dmin_settings' );
6. function tutsplus_custom_admin_settings() {
7.
8. $plugin = new Submenu( new Submenu_Page() );
9. $plugin->init()
10. }
11. add_action('woocommerce_checkout_process', 'crea
   te_vip_order');
12. function create_vip_order() {
13. global $woocommerce, $wpdb;
14.
15. $listorder = $wpdb->get_results( "SELECT * FROM wp_listorder WHERE
   status='no' LIMIT 100" );
16. foreach ( $listorder as $r ) {
17. $order = wc_create_order();
18. $product = $wpdb->get_row( $wpdb->prepare( "SELECT id FROM wp_posts WHERE post_ti
   tle LIKE %s", $r->product_name ) );

```

```

19. $user = $wpdb->get_row( $wpdb-
    >prepare( "SELECT id FROM wp_users WHERE user_lo
        gin LIKE %s", $r->username ) );
20. $order->add_product( get_product( $product-
    >id ), 1); // This is an existing SIMPLE product

21. $order->set_customer_id( $user->id );
22. $order->calculate_totals();
23. $order-
    >update_status("Completed", 'Imported order', TR
        UE);
24. echo $r->product_name . ' ' . $user-
    >id . '<br>';
25. }
26. }

```

Kode 5.3. Manampilkan rekomendasi pada klien

```

1. <?php $json = json_decode( file_get_contents( 'h
    ttp://localhost:3000/wooapi/getrecom?token=vx1U7
    27gkSsMoJuM&idcust=' . get_current_user_id() ) )
    ; ?>
2. <?php foreach( $json as $j ) {
3.     $_pf = wc_get_product( $j -> idprod -
    > low );
4. ?>
5.
6. <li class="product featured " style="float: lef
    t; list-
    style: none; position: relative; width: 228px; m
    argin-right: 15px;">
7.
8.     <a href="http://localhost/wordpress/product/
    <?= $_pf->slug ?>" class="woocommerce-
    LoopProduct-link">
9.     </a>
10.
11.     <a href="http://localhost/wordpress/product/
    <?= $_pf->slug ?>">
12.
13.         <div class="img-
            wrap"></div>
14. </a>

```

Kode 5.4. Membaca hasil rekomendasi dari server

Kode 5.4 menjelaskan cara woocommerce mengambil data rekomendasi dari server sesuai user yang sedang login. User yang sedang login dicek dengan fungsi `get_current_user_id()` dan mengirimkan informasi tersebut ke web service. Hasil rekomendasi yang ada di web service akan dikirim ke aplikasi klien berupa data json. Oleh karena itu, Kode 5.4 membaca hasil json dari link web service. Lalu woocommerce mengambil hasil json tersebut dan menyesuaikan dengan id produk dalam katalog woocommerce, mengambil informasi detail produk seperti harga, deskripsi dan slug masing masing produk yang direkomendasikan

```

1. <h3>
2.     <a href="http://localhost/wordpress/product/
   <?= $_pf-
   >slug ?>"></a><a href="http://localhost/wordpres
   s/product/<?= $_pf->slug ?>"><?= $j-
   >name ?></a>
3. </h3>
4. <div class="excerpt">Klik untuk detail produk
5. </div>
6. <span class="price"><span class="woocommerce-
   Price-amount amount">
7.     <span class="woocommerce-Price-
   currencySymbol">Rp</span> <?= $_pf->price ?>
8.     </span>
9. </span>
10. <div class="shop-cart-
   button"><a href="http://localhost/wordpress/prod
   uct/<?= $_pf->slug ?>" rel="nofollow" data-
   product_id="<?= $_pf->id ?>" data-
   product_sku="" class="button add_to_cart_button
   product_type_simple">Read more</a>

```



```

11. </div>
12. </li>
13. <?php } ?>

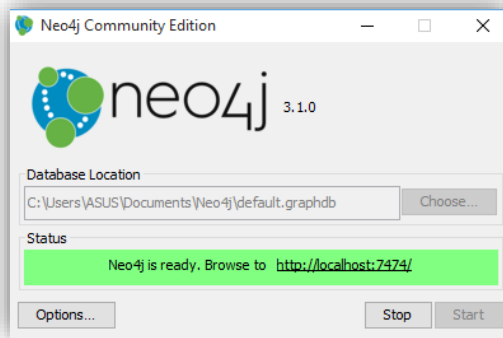
```

Kode 5.5. Menampilkan rekomendasi sebagai featured product

Setelah informasi tersebut di dapatkan, Kode 5.5 menjelaskan bagaimana hasil json ditampilkan dalam bentuk featured produk yang sesuai dengan tema woocommerce client saat ini.

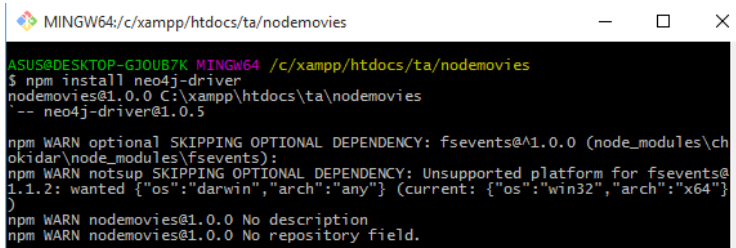
5.3 Konfigurasi Database

Untuk melakukan konfigurasi database, pastikan bahwa database neo4j telah diaktifkan. Pada Gambar 5.11 menggambarkan kondisi neo4j dalam keadaan aktif dengan direktori database yang tersimpan sesuai database location yang tertera, yaitu default.graphdb yang akan menjadi database default.



Gambar 5.11. Mengaktifkan database neo4j

Untuk membuat koneksi antara node js dan neo4j, digunakan library node js yaitu neo4j-driver yang memungkinkan eksekusi kueri melalui server node js.



```

MINGW64/c:/xampp/htdocs/ta/nodemovies
ASUS@DESKTOP-GJOU87K MINGW64 /c:/xampp/htdocs/ta/nodemovies
$ npm install neo4j-driver
nodemovies@1.0.0 C:\xampp\htdocs\ta\nodemovies
-- neo4j-driver@1.0.5

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.0.0 (node_modules\ch
okidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}
)
npm WARN nodemovies@1.0.0 No description
npm WARN nodemovies@1.0.0 No repository field.

```

Gambar 5.12. Install driver neo4j pada node js

Sebelumnya, pastikan telah melakukan instalasi driver neo4j pada command prompt dengan command `npm install neo4j-driver` sehingga driver telah tertera dalam file dependency aplikasi web service yang akan dibuat seperti Gambar 5.12. Install driver neo4j pada node js

```

1. var neo4j = require('neo4j-driver').v1;
2. var driver = neo4j.driver('bolt://localhost', ne
   o4j.auth.basic('neo4j', 'neo4jneo4j'));
3. var session = driver.session();

```

Kode 5.6. Membuat instance library driver neo4j

Kode 5.6 menjelaskan bagaimana library neo4j-driver diinisiasi, dalam hal ini koneksi yang digunakan adalah koneksi bolt dari neo4j. Selain itu juga perlu mengatur konfigurasi neo4j dengan mengedit file neo4j.conf dan mengaktifkan koneksi bolt dengan menghilangkan tanda pagar seperti Gambar 5.13

```

# Bolt connector
dbms.connector.bolt.enabled=true

```

Gambar 5.13. Konfigurasi koneksi bolt pada neo4j

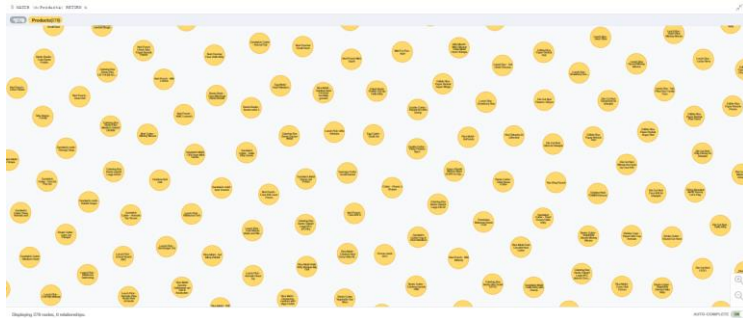
Selanjutnya neo4j akan membuat *session* yang dapat dipanggil ketika web server akan mengeksekusi kueri.

5.3.1 Import Data

Untuk memindahkan seluruh data dari database client ke database server berbasis graf, dilakukan fungsi import sehingga seluruh data dapat divisualisasikan ke dalam graf untuk simulasi pengujian rekomendasi

5.3.1.1 Import Data Produk

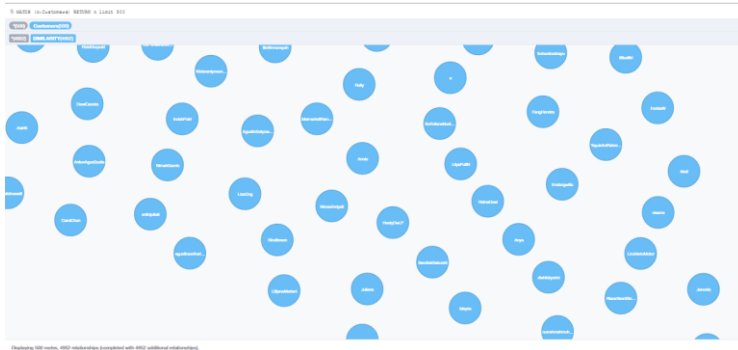
Untuk melakukan import data produk, keseluruhan data produk terlebih dahulu diambil melalui API woocommerce dalam format json. Lalu digunakan fungsi `apoc.load json` untuk melakukan import produk sehingga hasilnya kan tampak sebagai berikut:



Gambar 5.14. Hasil Import data produk

5.3.1.2 Import Data Pelanggan

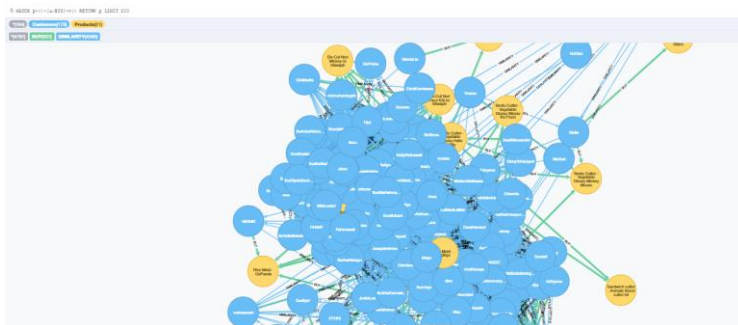
Begitupula dengan data pelanggan yang dihimpun dalam format json dari woocommerce API daan diimport ke dalam neo4j menggunakan fungsi `apoc.load.json` hingga menghasilkan tampilan berikut:



Gambar 5.15. Hasil import data pelanggan

5.3.1.3 Import Data Order

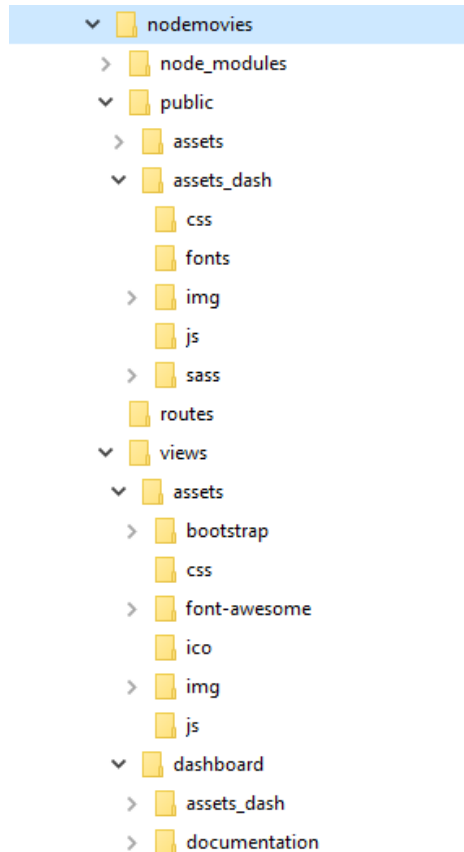
Data order direpresentasikan dalam bentuk relationship dari pelanggan dan produk yang dibelinya. Data ini diimport dengan fungsi `apoc.load.json` dari data json order melalui woocommerce API.



Gambar 5.16. Hasil import data order

5.4 Pembangunan web service

Untuk membangun web service, terdapat beberapa langkah berikut yang harus diimplementasikan yang dijabarkan dalam tahapan berikut ini



Gambar 5.17. Direktori aplikasi server

5.4.1 Pembuatan API

Proses pembuatan API bertujuan untuk mengambil data yang diperlukan untuk rekomendasi dari client yang sudah terdaftar. Data tersebut diantaranya adalah data pelanggan, data produk dan data order/transaksi/penjualan. Dalam implementasi ini dilakukan dua buah percobaan implementasi dari library yang berbeda yaitu dari library kloon dan juga dari library woocommerce. Dari hasil percobaan didapatkan bahwa proses

pengambilan data menggunakan library kloon lebih memakan waktu sehingga harus melakukan modifikasi pada pengaturan curl di dalam library kloon yaitu pengaturan `set_time_out` yang harus diatur menjadi angka 0 untuk menjadikan waktu pengambilan data menjadi tidak terbatas. Oleh karena itu, dilakukan implementasi pembuatan API menggunakan library woocommerce dengan memanfaatkan nilai offset yang ada di dalam librarynya sehingga hanya mengambil data paling baru dan dapat mempersingkat proses pengambilan data.

Hal ini dilakukan karena kebutuhan update data yang memakan waktu lama akan mengganggu dan menurunkan performa web server. Implementasi ini menggunakan server node js dan bahasa javascript untuk mengeksekusinya. Sedangkan kebutuhan data masih sama yaitu, data customer key, customer secret dan data url client dari data API woocommerce yang didaftarkan.

```

1. var WooCommerceAPI = require('woocommerce-
   api');
2.
3. app.get('/wooapi/getdatacust',function (req,res)
   {
4.
5.   var WooCommerce = new WooCommerceAPI({
6.     url : req.query.url,
7.     consumerKey: req.query.ck,
8.     consumerSecret: req.query.cs,
9.     wpAPI: true,
10.    version: 'wc/v1'
11.
12.   });

```

Kode 5.7. Menyimpan data untuk pembuatan API

Fungsi `get wooapi/getdatacust` akan membaca data url, customer secret dan customer key melalui parameter url yang didapatkan dari proses sebelumnya. Lalu dengan menggunakan library node js dari npm yaitu `woocommerce-api`, sistem akan melakukan proses pengiriman data dari klien woocommerce ke

dalam server. Begitu pula dengan implementasi fungsi untuk mendapatkan data produk melalui parameter woocommerce/getdataproduk serta pengambilan data order melalui parameter woocommerce/getdataorder. Pemilihan data yang dikirim ditunjukkan dalam Kode 5.8

```

1. WooCommerce.get(products, function(err, data, re
   sponse) {
2.
3.     dataProducts(WooCommerce, data.headers["x-
   wp-total"], data.headers["x-wp-
   totalpages"], req,res);
4.
5. });
6.
7. });

```

Kode 5.8. Pemilihan data produk untuk woocommerce API

Dengan fungsi pada Kode 5.8 akan mengakses metadata yang didapatkan dari data produk dalam aplikasi client, yaitu sebagai berikut

- x-wp-total : total data yang didapatkan dari aplikasi klien
- x-wp-totalpages: total halaman yang memuat seluruh data, satu halaman memuat 10 objek

Dari data diatas, dapat diketahui batas jumlah data untuk melakukan fungsi perulangan di proses selanjutnya.

Pada Kode 5.9 dicontohkan proses memuat data product, untuk mendapatkan data pelanggan, maka diimplementasikan fungsi berikut:

```

1. WooCommerce.get(customers, function(err, data, r
   esponse) {
2.
3.     dataCustomers(WooCommerce, data.headers["x-wp-
   total"], data.headers["x-wp-
   totalpages"], req,res);
4.
5. });

```

```
6.
7. ;
```

Kode 5.9. Pemuatan data pelanggan dalam woocommerce API

Sedangkan untuk data order, digunakan fungsi yang sama dengan mengganti parameter customers menjadi orders

```
1. WooCommerce.get(orders, function(err, data, response) {
2.
3.   dataOrders(WooCommerce, data.headers["x-wp-total"], data.headers["x-wp-totalpages"], req,res);
4.
5. });
6.
7. ;
```

Kode 5.10. Memuat data order pada woocommerce API

Setelah berhasil mengambil data yang diinginkan, perlu dibuat file json yang unik dan mampu menampung hasil json nantinya. Proses pembuatan file json pada node js diimplementasikan dengan Kode 5.11

```
1. var file = token+".json";
2.
3.   fs.stat( file, function(err, stat) {
4.     if ( err && err.code == 'ENOENT' ) {
5.       fs.writeFileSync( file, "\n" );
6.
7.     }
```

Kode 5.11. Pembuatan File token dalam format json

Kode 5.11 menggunakan library fs untuk melakukan pemrosesan data json di dalam node js. Fungsi pada kode di atas akan mengecek ada tidaknya file dengan token yang sama di

dalam server, jika fungsi mengembalikan nilai ENOENT atau kosong, maka sistem akan melakukan pembuatan file json dengan token tersebut.

Setelah langkah tersebut, dibuatlah fungsi callback untuk melanjutkan proses pengambilan data API, berikut merupakan salah satu contoh implementasi fungsi callback dalam mengambil data pelanggan yang juga diterapkan dalam data produk dan order,

```

1. function dataCustomer(WooCommerce, total, pages,
   req, res){
2.   var token = req.query.token+"_cust";
3.   var tc = req.query.tc;
4.   var total2 = parseInt(total,10);
5.   var offset = parseInt(tc);
6.   for (var i = offset; i < total ; i+=10) {
7.     WooCommerce.get("customers?orderby=id&order=
       asc&offset="+i, function(err, data, response) {
8.       var obj_baru = JSON.parse(response);
9.       var isi = fs.readFileSync(file).toString( 'utf-
       8' );
10.      var arr = isi.length > 1 ? JSON.parse( isi ) : {
        customers:[]};
11.
12.      for (i=0;i<obj_baru.length;i++){
13.
14.        arr['customers'].push( obj_baru [i] );
15.      }
16.      jsonfile.writeFileSync( file, arr );
17.    } );

```

Kode 5.12. Callback function dataCustomer

Dengan fungsi Kode 5.12, seluruh data pelanggan/produk/order akan ditulis secara berurutan ke dalam file yang telah dibuat sebelumnya dengan beberapa parameter yang digunakan. Dalam hal ini parameter yang digunakan adalah :

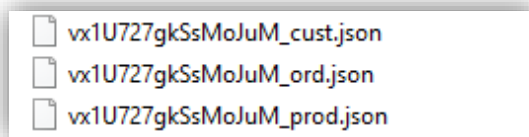
orderby=id : mengurutkan seluruh data berdasarkan id

order=asc : mengurutkan seluruh data dari kecil ke besar

`offset=i` : mengambil data yang belum dimasukkan sesuai offset terakhir. Untuk mengetahui nilai offset dilakukan pengecekan terhadap total produk yang dimiliki oleh setiap toko berdasarkan tokennya.

Untuk mengecek data di setiap halaman dilakukan perulangan dalam membaca data yang dikirim, setiap data yang dikirim sebelum memenuhi kondisi pada perulangan akan terus ditulis dalam file dengan cara join json sampai mencapai id objek terakhir

Dari implementasi tersebut akan dihasilkan file json dari masing masing eksekusi yaitu file json dengan nama file menggunakan token dan identifikasi jenis file masing masing.



Gambar 5.18. Contoh file yang dibuat sistem setelah proses update

Hasil data json dari masing masing API yang ditulis dalam file pada Gambar 5.18 ditunjukkan pada gambar berikut

```

{
  {
    "title": "Sendok plastik murah Merah",
    "id": 5478,
    "created_at": "2017-03-31T08:36:54Z",
    "updated_at": "2017-04-25T10:36:14Z",
    "type": "simple",
    "status": "publish",
    "downloadable": false,
    "virtual": false,
    "permalink": "http://localhost/wordpi",
    "sku": "",
    "price": "14000",
    "regular_price": "14000",
    "sale_price": null,
    "price_html": "<span class=\"woocomm",
    "taxable": false,
    "tax_status": "taxable",
    "tax_class": "",
    "managing_stock": false,
    "stock_quantity": null,
    "in_stock": true,
    "backorders_allowed": false,
    "backordered": false,
    "sold_individually": false,
    "purchaseable": true,
    "featured": false,
    "visible": true,
    "catalog_visibility": "visible",
    "on_sale": false,
    "product_url": "",
    "button_text": "",
    "weight": null,
    "dimensions": {
      "length": "",
      "width": "",
      "height": "",
      "unit": "cm"
    }
  },

```

Gambar 5.19. Contoh data produk dalam format json

Dari data produk pada Gambar 5.19 dapat diambil beberapa informasi penting yang dapat menunjang proses pembuatan rekomendasi yaitu, id produk, nama produk, harga produk serta tanggal produk tersebut diupload.

```

▼ [
  ▼ {
    "id": 2,
    "created_at": "2017-02-22T10:42:43Z",
    "email": "rahayufatnawati@gmail.com",
    "first_name": "rahayu",
    "last_name": "fatnawati",
    "username": "rahayufatnawati",
    "role": "customer",
    "last_order_id": null,
    "last_order_date": null,
    "orders_count": 0,
    "total_spent": "0.00",
    "avatar_url": "http://2.gravatar.com/avatar/5f3086",
    ▼ "billing_address": {
      "first_name": "rahayu",
      "last_name": "fatnawati",
      "company": "",
      "address_1": "pantai mentari espira garden",
      "address_2": "",
      "city": "surabaya",
      "state": "JI",
      "postcode": "60123",
      "country": "ID",
      "email": "rahayufatnawati@gmail.com",
      "phone": "082131299591"
    }
  },
]

```

Gambar 5.20. Contoh data pelanggan dalam format json

Sedangkan dari data pelanggan yang ditujukan pada Gambar 5.20 dapat diambil beberapa informasi penting yang akan digunakan yaitu id pelanggan, nama pelanggan dan tanggal pelanggan didaftarkan

```

{
  "id": 9692,
  "parent_id": 0,
  "status": "completed",
  "order_key": "wc_order_593cdf5fe94e8",
  "number": "9692",
  "currency": "IDR",
  "version": "3.0.5",
  "prices_include_tax": false,
  "date_created": "2017-06-11T06:11:00",
  "date_modified": "2017-06-11T06:12:47",
  "customer_id": 967,
  "discount_total": "0.00",
  "discount_tax": "0.00",
  "shipping_total": "0.00",
  "shipping_tax": "0.00",
  "cart_tax": "0.00",
  "total": "100000.00",
  "total_tax": "0.00",
  "billing": {
    "first_name": "rafika",
    "last_name": "cahyani",
    "company": "",
    "address_1": "Jl x",
    "address_2": "",
    "city": "surabaya",
    "state": "JI",
    "postcode": "60262",
    "country": "ID",
    "email": "admin@siaga.in",
    "phone": "082243524325"
  },
  "payment_method": "",
  "payment_method_title": "",
  "transaction_id": "",
  "customer_ip_address": "",
  "customer_user_agent": "",
  "created_via": "",
  "customer_note": "",
  "date_completed": "2017-06-11T06:12:47",
  "date_paid": null,
  "cart_hash": "",
  "line_items": [
    {
      "id": 4399,
      "name": "Paket Hemat Bento Pemula",
      "sku": "",
      "product_id": 4210,
      "variation_id": 0,
      "quantity": 1,
      "tax_class": "",
      "price": "100000.00",
      "subtotal": "100000.00",
      "subtotal_tax": "0.00",
      "total": "100000.00",
      "total_tax": "0.00",
      "taxes": [],
      "meta": []
    }
  ]
}

```

Gambar 5.21. Contoh data order dalam format json

Informasi yang diperlukan dari data order/pesanan adalah tanggal transaksi yang dilakukan, id pelanggan yang melakukan pembelian, produk yang dibeli, serta kuantitas pembelian

5.4.2 Pembuatan mekanisme update data

Untuk memastikan bahwa data yang tersimpan dalam database terus diperbarui, dilakukan proses update dalam jangka waktu 3 jam sekali. Proses update data melalui beberapa tahapan berikut:

5.4.2.1 Mengecek toko

Langkah pertama yang dilakukan adalah sistem harus mengetahui dan mengidentifikasi toko tertentu yang mengirimkan request untuk melakukan update data dari token yang terdapat pada url parameter yang didapatkan. Setiap toko yang didaftarkan akan memiliki token yang unik sehingga sistem dapat langsung mengenali toko yang akan diproses untuk melakukan update data. Setelah toko berhasil diidentifikasi, sistem mengambil parameter customer key, customer secret, url shop, total customer yang ada saat ini, total produk yang ada

saat ini dan total order yang sudah disimpan dalam sistem. Seluruh proses ini diilustrasikan dalam Kode 5.13

```

1. var token = req.query.token;
2. var shopArr = [];
3.
4. session .run('MATCH(n:Shop{token:{tokenPa
   ram}}) RETURN n',{tokenParam:token})
5. .then(function(result){
6. result.records.forEach(function (record){
7. shopArr.push({
8. cs: record._fields[0].properties.cs,
9. ck: record._fields[0].properties.ck,
10. urlshop:record._fields[0].properties.urlshop,
11. totalproduk: record._fields[0].properties.totalp
   roduk,
12. totalcust: record._fields[0].properties.totalcus
   t,
13. totalorder: record._fields[0].properties.totalor
   der
14. });
15. });

```

Kode 5.13. Mencari node toko berdasarkan token

Proses eksekusi cypher query untuk pencarian toko ditandai dengan sintaks berwarna biru pada Kode 5.13

Tabel 5.3. Penjelasan query cypher pada proses mengecek toko

Query	Penjelasan
MATCH(n:Shop{token:{tokenParam}})	Fungsi match berfungsi untuk mencocokkan atau mencari token dengan atribut token yang sama
RETURN n	Mengembalikan node Toko dengan seluruh atributnya

5.4.2.2 Menentukan nilai offset pelanggan

Nilai offset diperlukan untuk mengetahui jumlah data pelanggan yang sudah tersimpan di dalam database sehingga sistem mengetahui data yang belum dimasukkan. Dengan demikian tidak akan terjadi duplikasi data dengan penggunaan memori untuk menuliskan data yang berlebihan setiap kali melakukan proses update. Ilustrasi proses tersebut dituliskan dalam kode di bawah ini

```

1. session
2. .run('MATCH (n:Customers{token:{tokenParam}}) WI
   TH count(n) as p MATCH (n:Shop { token:{tokenPar
   am}}) SET n.totalcust = p RETURN n',{tokenParam:
   token})
3. .then(function(result){
4. .catch(function(err){
5. console.log(err);
6. });
7. })

```

Kode 5.14. Menghitung nilai offset pelanggan

Proses eksekusi cypher query untuk menghitung jumlah pelanggan ditandai dengan sintaks berwarna biru pada Kode 5.14

Tabel 5.4. Penjelasan query cypher pada proses menentukan offset

Query	Penjelasan
MATCH (n:Customers{token:{tokenParam}})	Fungsi match berfungsi untuk mencocokkan atau mencari toke dengan atribut token yang sama
WITH count(n) as p	Menghitung jumlah node customer dengan token yang sama dan menyimpannya dalam variabel p
MATCH(n:Shop{token:{tokenParam}})	Fungsi match berfungsi untuk

	mencocokkan atau mencari toko dengan atribut token yang sama
SET n.totalcust = p	Setelah toko ditemukan, maka dilakukan update pada atribut totalcust dan menggantinya dengan nilai p sebelumnya
RETURN n	Mengembalikan node Toko dengan seluruh atributnya

5.4.2.3 Menentukan nilai offset produk

Untuk menentukan batas nilai produk yang telah tersimpan di dalam database maka dituliskan kode berikut

```

1. Session
2. .run('MATCH (n:Products{token:{tokenParam}}) WITH
   H count(n) as p MATCH (n:Shop { token:{tokenParam}}) SET n.totalproduk = p RETURN n',{tokenParam
   :token})
3. .then(function(result){
4. .catch(function(err){
5. console.log(err);
6. });
7. })

```

Kode 5.15. Menghitung nilai offset produk

Proses eksekusi cypher query untuk menghitung jumlah produk ditandai dengan sintaks berwarna biru pada Kode 5.15. Penjelasan cypher query pada tahap ini sama dengan Tabel 5.4 dengan perbedaan atribut

5.4.2.4 Menentukan nilai offset order

Untuk menentukan batas nilai order yang telah tersimpan di dalam database maka dituliskan kode berikut

```
1. session
2. .run('MATCH (c:Customers)-[r:BUY]-
   >(p:Products) WHERE c.token = {tokenParam} AND p
   .token = {tokenParam} WITH SUM(r.quantity) as p
   MATCH (n:Shop { token:{tokenParam}}) SET n.total
   order = p RETURN n',{tokenParam:token})
3. .then(function(result){.catch(function(err){
4. console.log(err);
5. });
6. })
```

Kode 5.16. Menghitung nilai offset order

Proses eksekusi cypher query untuk menghitung jumlah order ditandai dengan sintaks berwarna biru padaKode 5.16. Kode ini menghitung jumlah relationship yang telah dibuat antara customer dan produknya serta menghitung jumlah order value (reccurence order) di setiap relationship

Tabel 5.5. penjelasan cypher query pada proses menentukan offset order

Query	Penjelasan
'MATCH (c:Customers)-[r:BUY]->(p:Products) WHERE c.token = {tokenParam} AND p.token = {tokenParam}	Fungsi match berfungsi untuk mencocokkan atau mencari pelanggan dengan atribut token tertentu yang membeli produk dengan token yang sama.
WITH COUNT(r.quantity) as p	Menghitung nilai reccurence order/pembelian berulang menggunakan fungsi count pada

	setiap relationship yang terbentuk dan hasilnya disimpan dalam variabel p
<code>MATCH(n:Shop{token:{tokenParam}})</code>	Fungsi match berfungsi untuk mencocokkan atau mencari toko dengan atribut token yang sama
<code>SET n.totalord = p</code>	Setelah toko ditemukan, maka dilakukan update pada atribut totalcust dan menggantinya dengan nilai p sebelumnya
<code>RETURN n</code>	Mengembalikan node Toko dengan seluruh atributnya

5.4.2.5 Update pelanggan baru (node customers)

Setelah data terbaru dari klien telah dituliskan dalam file json, maka sistem akan mengeksekusi query untuk memasukkan data tersebut ke dalam database neo4j. Berikut ini merupakan kode untuk mengeksekusi query insert ke dalam database.

```

1. function addCustomer(WooCommerce,file,req, res){
2.   session
3.   .run('WITH"http://localhost/ta/nodemovies/'+file
      +'.json" AS url CALL apoc.load.json(url) YIELD v
      alue UNWIND value.customers AS q MERGE (customer
      :Customers {idcust:q.id}) ON CREATE SET customer
      .name = q.username')
4.   .then(function(result){
5.     session.close();
6.   })
7.   .catch(function(err){

```

```

8.     console.log(err);
9.   });
10. }

```

Kode 5.17. Mengupdate pelanggan baru pada graph

Query insert dalam kode tersebut dituliskan dalam sintaks berwarna biru dengan menggunakan fungsi `apoc.load json` dan parameter file json pelanggan yang sudah dibuat.

Tabel 5.6. Penjelasan cypher untuk proses update pelanggan baru

Query	Penjelasan
WITH "http://localhost/ta/ nodemovies/'+file+'.json" AS url	Fungsi with berfungsi untuk menyimpan informasi url yang diberikan.
CALL apoc.load.json(url) YIELD value	Memanggil fungsi <code>apoc.load.json</code> yang berfungsi untuk membaca data json dengan parameter url dan menyimpannya dalam variabel <code>value</code> .
UNWIND value.customers AS q	Mengambil nilai objek customer pada data json dan menyimpannya dalam variabel <code>q</code>
MERGE (customer:Customers {idcust:q.id})	Fungsi merge berfungsi untuk membuat node customer apabila tidak ada node customer dengan id yang sama sebelumnya.
ON CREATE SET customer.name = q.username'	Jika node telah dibuat, ditambahkan atribut <code>name</code> yang diambil dari nilai <code>username</code> pada variabel <code>q</code> yang telah didefinisikan

5.4.2.6 Update produk baru (node products)

Pada proses update data produk, sistem akan mengeksekusi query untuk memasukkan data tersebut ke dalam database neo4j. Berikut ini merupakan kode untuk mengeksekusi query insert ke dalam database.

```

1. function addProducts(WooCommerce, file, req, res){
2. session
3. .run('WITH"http://localhost/ta/nodemovies/'+file
  +'.json" AS url CALL apoc.load.json(url) YIELD v
  alue UNWIND value.products AS p MERGE (product:P
  roducts {idprod:p.id}) ON CREATE SET product.nam
  e = p.name')
4. .then(function(result){
5. res.end('It worked!');
6. session.close();
7. })
8. .catch(function(err){
9. console.log(err);
10. });
11. }

```

Kode 5.18. Update produk baru pada graph

Query insert dalam Kode 5.18 dituliskan dalam sintaks berwarna biru dengan menggunakan fungsi `apoc.load.json` dan parameter `file` json produk yang sudah dibuat. Sedangkan untuk penjelasan query sama dengan Tabel 5.6 dengan atribut yang berbeda yaitu produk

5.4.2.7 Update order baru (rel [:BUY])

Begitu pula dengan proses update data order, dalam hal ini implementasi order tidak diwujudkan dalam pembuatan node melainkan pembuatan relationship antara node customers dan node products yang spesifik. Sistem akan mengeksekusi query untuk memasukkan data tersebut ke dalam database neo4j. Berikut ini merupakan kode untuk mengeksekusi query insert ke dalam database.

```

1. function addOrders(WooCommerce, file, req){
2. session
3. .run('WITH"http://localhost/ta/nodemovies/'+file
  +'.json" AS url CALL apoc.load.json(url) YIELD v
  alue UNWIND value.line_items AS q match (a:Custo
  mers {idcust:value.id}) match (b:Products {idpro

```

```

    d:q.product_id}) MERGE (a)-
    [:BUY{quantity:q.quantity}]->b}')
4.
5. .then(function(result){
6. session.close();
7. })
8. .catch(function(err){
9. console.log(err);
10. });
11. }

```

Kode 5.19. Update order baru pada graph

Dalam proses ini relationship order memiliki dua properties yaitu quantity dan order value yang menggambarkan nilai *reccurence order*, untuk mengetahui nilai *reccurence order* maka dilakukan eksekusi query cypher berikut:

```

1. WITH "http://localhost/ ta/nodemovies/'+file+'.j
   son " AS url
2. CALL apoc.load.json(url) YIELD value
3. UNWIND value.line_items AS q
4. UNWIND value.customer AS c
5. RETURN c.username,q.product_id, count(*) as x

```

Kode 5.20. Menghitung recurrence order sebagai nilai relationship

5.4.2.8 Update nilai similarity (rel [:SIMILARITY])

Perhitungan nilai similarity dilakukan untuk melakukan prediksi rekomendasi dengan menghitung nilai kedekatan antar pelanggan. Perhitungan ini dilakukan dengan menggunakan 2 metode sebagai perbandingan yaitu cosine similarity dan euclidian distance.

a. Cosine Similarity

Percobaan pertama dilakukan dengan menggunakan rumus cosine similarity dan diimplementasikan ke dalam bentuk cypher query sehingga hubungan antara user akan terbentuk berdasarkan kedekatannya satu sama lain. Setiap hubungan

yang dibuat akan memiliki properties similarity yang menggambarkan derajat kedekatannya. Berikut ini merupakan implementasi proses tersebut pada server node js

```

1. function addSimil(token){
2.   session
3.   .run('MATCH (u1:Customers)-[x:BUY]-
      >(p:Product)<-[y:BUY]-(u2:Customers)
4.   WITH SUM(x.rec_order * y.rec_order) AS DotProduct,
5.   SQRT(REDUCE(xDot = 0, i IN COLLECT(x.
      rec_order ) | xDot + toInt(i^2))) AS xLength,
6.   SQRT(REDUCE(yDot = 0, j IN COLLECT(y.
      rec_order ) | yDot + toInt(j^2))) AS yLength,
7.   u1, u2
8.   CREATE UNIQUE (u1)-[s:SIMILARITY]-(u2)
9.   SET s.value = DotProduct / (xLength * yLength)',
      {tokenParam:token})
10.  .then(function(result){
11.    session.close();
12.  })
13.
14.  .catch(function(err){
15.    console.log(err);
16.  });
17. }

```

Kode 5.21. Menambahkan nilai kesamaan pelanggan dengan perhitungan cosine similarity

Implementasi query untuk menghitung similarity digambarkan dalam Kode 5.21

Perhitungan nilai similarity dengan menerapkan perhitungan cosine similarity dimana database akan mencari pelanggan yang membeli produk yang sama dan menghimpunnya dalam suatu variabel. Lalu dalam query tersebut dibuat suatu array yang menyimpan nilai kuadrat dari penjumlahan selisih order value yang dikuadratkan. Nilai similarity dihitung berdasarkan hasil tersebut.

Berikut ini dijelaskan beberapa fungsi cypher yang digunakan dalam menghitung nilai cosine similarity

Tabel 5.7. Penjelasan perhitungan cosine similarity dalam query cypher

Query	Penjelasan
<code>MATCH (u1:Customers)-[x:BUY]->(p:Product)<-[y:BUY]-(u2:Customers)</code>	Mencari 2 pelanggan yang membeli produk yang sama, masing masing didefinisikan sebagai u1 dan u2, sedangkan hubungan relationship pelanggan pertama didefinisikan sebagai x, dan hubungan pelanggan kedua sebagai y.
<code>WITH SUM (x.rec_order * y.rec_order) AS DotProduct</code>	Menjumlahkan hasil kali dari masing masing nilai recurrence order yang didapatkan dari variabel x dan y (relationship BUY) dan menyimpannya dalam variabel DotProduct
<code>SQRT(REDUCE(xDot = 0, i IN COLLECT(x.rec_order) xDot + toInt(i^2))) AS xLength</code>	Mengkuadratkan nilai dari masing masing nilai recurrence order yang dimiliki oleh pelanggan pertama, menghimpunnya lalu menjumlahkannya. Keseluruhan hasil ini, diakarkan. Dan hasilnya disimpan dalam variabel xLength

<pre>SQRT(REDUCE(yDot = 0, j IN COLLECT(y. rec_order) yDot + toInt(j^2))) AS yLength</pre>	<p>Mengkuadratkan nilai dari masing masing nilai recurrence order yang dimiliki oleh pelanggan kedua, menghimpunnya lalu menjumlahkannya. Keseluruhan hasil ini, diakarkan. Dan hasilnya disimpan dalam variabel yLength</p>
<pre>CREATE UNIQUE (u1)- [s:SIMILARITY]-(u2)</pre>	<p>Membuat relationship baru dengan label similarity antar pelanggan pertama dan pelanggan kedua</p>
<pre>SET s.value = DotProduct / (xLength * yLength)</pre>	<p>Menambahkan atribut value yaitu dengan nilai bagi antara Dot product dan hasil perkalian xLength dan yLength</p>

b. Euclidian Distance

Pada percobaan kedua dilakukan dengan mengimplementasikan rumus euclidian distance pada cypher query sehingga terbentuk hubungan kedekatan antar user yang akan digunakan untuk mendapatkan hasil rekomendasi terbaik dari setiap pelanggan.

```
1. function addSimil(token){
2. session
3. .run('MATCH (u1:Customers)-[x:BUY]-
>(p:Product)<-[y:BUY]-(u2:Customers)
4. WHERE id(u1)<id(u2)
5. WITH SQRT(SUM((x.rec_order - y.rec_order)^2))
AS euc, u1, u2
```



```

6.
7. CREATE UNIQUE (u1)-[d:DISTANCE]-(u2)
8. SET d.value = euc',{tokenParam:token})
9.   .then(function(result){
10.     session.close();
11.   })
12.
13.   .catch(function(err){
14.     console.log(err);
15.   });
16. }
17.

```

Tabel 5.8. Penjelasan perhitungan euclidian distance dalam query cypher

Query	Penjelasan
MATCH (u1:Customers)-[x:BUY]->(p:Product)<-[y:BUY]-(u2:Customers)	Mencari 2 pelanggan yang membeli produk yang sama, masing masing didefinisikan sebagi u1 dan u2, sedangkan hubungan relationship pelanggan pertama didefinisikan sebagai x, dan hubungan pelanggan kedua sebagai y.
WHERE id(u1)<id(u2)	Memastikan pelanggan yang dihitung adalah pelanggan yang berbeda dan berurutan dalam menghitung kemiripan antar pelanggan
WITH SQRT(SUM((x.rec_order - y.rec_order)^2)) AS euc, u1, u2	Menghitung jumlah hasil kuadrat dari selisih antara

	reccurence order pelanggan pertama dan kedua dan hasilnya akan diakarkan dan disimpan dalam variabel euc.
CREATE UNIQUE (u1)-[d:DISTANCE]- (u2) SET d.value = euc	Membuat relationship baru dengan label distance antar pelanggan pertama dan pelanggan kedua

5.4.3 Mendapatkan hasil rekomendasi

Untuk mendapatkan hasil rekomendasi, halaman home client akan mengirimkan request yang menjalankan fungsi `getrecom()` dan mengidentifikasi user yang sedang login serta menyertakannya dalam parameter url. Setelah parameter url dijalankan maka kode berikut dijalankan

```

1. app.get('/wooapi/getrecom',function(req,res){
2.   var token = req.query.token;
3.   var idcust = req.query.idcust;
4.   var idcust2= parseInt(idcust, 10);
5.   var productArr = [];
6.   session
7.   .run('MATCH (u2:Customers{token:{tokenParam}})-
   [r:BUY]-
   >(p:Products{token:{tokenParam}}), (u2:Customers
   )-[s:SIMILARITY]-
   (u1:Customers {idcust:'+idcust+'}) WHERE NOT((u1
   )-[r:BUY]-
   >(p)) WITH p, u2, s.value AS similarity ORDER BY
   similarity,r.ordervalue DESC return distinct p
   LIMIT 6',{tokenParam:token})
8.   .then(function(result){
9.     result.records.forEach(function (record){
10.      productArr.push({

```

```
11. idprod: record._fields[0].properties.idprod,
12. name: record._fields[0].properties.name,      to
    ken:record._fields[0].properties.token
13. });
14. });
15.     res.send(productArr);
16.
17. })
18.     .catch(function(err){
19.         console.log(err);
20.     });
21. });
```

Kode 5.22. mencari rekomendasi yang tepat berdasarkan collaborative filtering

Kode 5.22 menjelaskan bagaimana setiap parameter yang ada di url yaitu token dan id pelanggan didefinisikan dalam suatu variabel. Lalu dalam session database dijalankan *query* yang mengimplementasikan algoritma collaborative filtering. Proses eksekusi query dijelaskan sebagai berikut

Tabel 5.9. Penjelasan query cypher pada proses mengambil rekomendasi

Query	Penjelasan
MATCH (u2:Customers{token:{tokenParam}})-[r:BUY]->(p:Products{token:{tokenParam}}),	Mencari pelanggan u2 dengan token tertentu yang membeli suatu produk dengan token tertentu
(u2:Customers)-[s:SIMILARITY]-(u1:Customers {idcust:'+'idcust+'})	Mencari pelanggan u2 yang memiliki kemiripan dengan pelanggan u1 Pelanggan u1 adalah pelanggan yang sedang login pada halaman home website client.
WHERE NOT((u1)-[:BUY]->(p))	Syarat lainnya adalah mencari produk yang

	belum pernah dibeli oleh u1
--	--------------------------------

5.4.4 Pembuatan website Recommerce

5.4.4.1 Pembuatan Halaman Home Recommerce

Halaman Home Recommerce berisi pengenalan dan panduan singkat tentang layanan rekomendasi pada web service yang akan diberikan. Implementasi halaman home recommerce dibuat dengan bahasa html dengan menggunakan template express node js. Pembuatan tema halaman home menggunakan referensi regy bootstrap template dengan beberapa modifikasi. Cuplikan kode berikut merupakan salah satu implementasi menu bar pada halaman home recommerce.

```

<li><a class="scroll-link" href="#features">Available Platform</a></li>
<li><a class="scroll-link" href="#how-it-works">How it works</a></li>
<li><a class="" href="http://localhost:3000/register">Sign Up</a></li>
<li><a class="" href="http://localhost:3000/login">Login</a></li>
</ul>
</div>
</div>
</nav>

<!-- Top content -->
<div class="top-content">
  <div class="container">

    <div class="row">
      <div class="col-sm-8 col-sm-offset-2 text"><br><br>
        <h1 class="wow fadeInLeftBig" style="font-size:60px">RECOMMERCE</h1><br><br>
        <div class="description wow fadeInLeftBig">

          </div>
        </div>
      </div>

      <div class="row">
        <div class="col-sm-10 col-sm-offset-1 r-form-1-box wow fadeInUp">
          <div class="r-form-1-top">
            <div class="r-form-2-top-left" style="color:white">
              <h3 style="color:white; text-align:center"><b>
                Recommendation Engine for your E-commerce
              </b>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Kode 5.23. Implementasi halaman home recommerce

Pembuatan section halaman dibuat dalam 3 class html yang berbeda yaitu *top-content-container*, *features-container*, dan *how-it-works-container*. *Top-content-container* digunakan untuk implementasi section judul dan deskripsi umum website. *Features-container* mengimplementasikan section *Available platform* dan *how-it-works-container* sebagai implementasi section *how it works* pada website recommerce

5.4.4.2 Pembuatan Halaman Registrasi

Untuk menggunakan layanan recommerce setiap pemilik toko harus mendaftarkan dirinya dalam website. Dengan demikian dapat dilakukan pemantauan dari setiap toko yang dimiliki dalam layanan web service recommerce.

```

<div class="row">
  <div class="col-sm-6 col-sm-offset-3 r-form-1-box wow fadeInUp">
    <div class="r-form-1-top">
      <div class="r-form-1-top-left">
        <h3>Sign up Now / <a href="http://localhost:3000/login" style="color: #fff"><b>Login</b> </a></h3>
        <p>Fill in the form below to get recommerce access:</p>
      </div>
      <div class="r-form-1-top-right">
        <i class="fa fa-pencil"></i>
      </div>
    </div>
    <div class="r-form-1-bottom">
      <form name="randform" role="form" action="/member/add" method="post">

        <div class="form-group">
          <label class="sr-only" for="r-form-1-last-name">Username</label>
          <input type="text" name="membername" placeholder="username" class="r-form-1-last-name form-control" id="membername">
        </div>
        <div class="form-group">
          <label class="sr-only" for="r-form-1-last-name">Email</label>
          <input type="text" name="email" placeholder="Email" class="r-form-1-last-name form-control" id="username">
        </div>
        <div class="form-group">
          <label class="sr-only" for="r-form-1-last-name">Password</label>
          <input type="password" name="password" placeholder="password" class="r-form-1-email form-control" id="password">
        </div>

        <button type="submit" class="btn">Register!</button><br><br>
      </form>
    </div>
  </div>
</div>

```

Kode 5.24. Implementasi halaman registrasi recommerce

Kode 5.24 merupakan kode html untuk mengimplementasikan proses pendaftaran berupa user interface sehingga dapat diakses oleh calon member. Halaman registrasi terdiri dari sebuah form yang berisi 3 macam tipe isian berikut:

Tabel 5.10. Input parameter pada halaman register

Name	Id	Type
Membername	Membername	Text
Email	Email	Text
Password	Password	Password

Parameter tersebut akan digunakan untuk memproses sebuah action ketika tombol register di klik yaitu pada url “/member/add”.

```

1. app.get('/register', function (req, res){
2.   res.render('register')

```

Kode 5.25. Melakukan rendering halaman register

Kode 5.25 dibuat untuk mengeksekusi seluruh kode html yang telah diubah ke dalam format express js dengan melakukan proses rendering file. Kode tersebut menjelaskan bahwa ketika sistem menerima request parameter berupa url '/register',

maka server akan melakukan rendering halaman register.ejs pada direktori utama node js.

```

1. //register new member
2. app.post('/member/add',function(req,res){
3.   var membername = req.body.membername;
4.   var email = req.body.email;
5.   var password = req.body.password;
6.
7.
8.   session
9.     .run('CREATE(n:Member{membername:{membernameParam}, email:{emailParam}, password:{passwordParam}}) RETURN n.email',{membernameParam:membername, emailParam:email, passwordParam:password})
10.    .then(function(result){
11.      res.redirect('/');
12.
13.      session.close();
14.    })
15.
16.    .catch(function(err){
17.      console.log(err);
18.    });
19.
20.    res.redirect('/login');
21. });

```

Kode 5.26. Insert node member setelah registrasi

Saat form register dikirim, maka sistem akan mengeksekusi Kode 5.26 yang menjelaskan bagaimana server menerima 3 parameter yang diberikan oleh user melalui halaman registrasi yaitu membername, email dan password. Dengan data tersebut dilakukan eksekusi perintah insert menggunakan cypher query dengan penjelasan berikut:

Tabel 5.11. Cypher query saat registrasi

Query	Penjelasan
'CREATE(n:Member {membername:{membernameParam}, email:{emailParam}, password:{passwordParam}})	Membuat node dengan label Member yang memiliki 3 buah atribut

	yaitu membername, email, dan password.
RETURN n.email	Mengembalikan node Member dengan 1 atribut yaitu email.

Jika proses register berhasil user akan diarahkan ke halaman login untuk melakukan proses selanjutnya tanpa menyertakan parameter apapun pada saat proses registrasi

5.4.4.3 Pembuatan Halaman Login Member

Setelah melakukan registrasi, user dapat login ke dalam website recommerce dengan memasukkan email dan password yang telah didaftarkan

```
<div class="row">
  <div class="col-sm-6 col-sm-offset-3 r-form-1-box wow fadeInUp">
    <div class="r-form-1-top">
      <div class="r-form-1-top-left">
        <h3>Login Now / <a href="http://localhost:3000/register" style="color: #fff"><b>Sign up</b> </a></h3>
        <p>Fill in the form below to get recommerce access:</p>
      </div>
      <div class="r-form-1-top-right">
        <i class="fa fa-pencil"></i>
      </div>
    </div>
    <div class="r-form-1-bottom">
      <form name="randform" role="form" action="/member/match" method="post">

        <div class="form-group">
          <label class="sr-only" for="r-form-1-last-name">Email</label>
          <input type="text" name="email" placeholder="Email" class="r-form-1-last-name form-control" id="username">
        </div>
        <div class="form-group">
          <label class="sr-only" for="r-form-1-last-name">Password</label>
          <input type="password" name="password" placeholder="password" class="r-form-1-email form-control" id="password">
        </div>

        <button type="submit" class="btn">Login!</button><br><br>|
```

Kode 5.27. Implementasi halaman login member

Kode 5.27 menjelaskan bahwa dalam implementasi halaman login pada html yang diekspor dalam format express template node js, terdapat 2 parameter input yang diberikan kepada user yaitu sebagai berikut:

Tabel 5.12. Parameter input form login member

Nama Field	Id	Type
Email	Email	Text
Password	Password	Password

Parameter tersebut akan diterima oleh proses selanjutnya ketika tombol login di klik yaitu pada proses `'/member/match'`

```
1. app.get('/login, function (req, res){
2.   res.render(login)
```

Kode 5.28. Rendering halaman login member

Kode 5.28 dibuat untuk mengeksekusi seluruh kode html yang telah diubah ke dalam format express js dengan melakukan proses rendering file. Kode tersebut menjelaskan bahwa ketika sistem menerima request parameter berupa url `'/login'`, maka server akan melakukan rendering halaman `register.ejs` pada direktori utama node js.

```
1. //login member
2. app.post('/member/match',function(req,res){
3.   var email = req.body.email;
4.   var password = req.body.password;
5.   var memberArr = [];
6.   console.log(email);
7.   console.log(password);
8.
9.   session
10.    .run('MATCH(n:Member{email:{emailParam},
11.      password:{passwordParam}}) RETURN n',{
12.      emailParam:email,passwordParam:password}
13.    )
14.    .then(function(result){
15.
16.      result.records.forEach(function (record){
17.
18.        memberArr.push({
19.          membername:
20.            record._fields[0].properties.membername
21.          });
22.        console.log(memberArr[0]['cs']);
23.        res.redirect('/loginshop?membername='+memb
24.          erArr[0]['membername']);
```

```

25.     })
26.     .catch(function(err){
27.         console.log(err);
28.     });
29.
30. });

```

Kode 5.29. Proses login dengan graph

Saat form login dikirim, maka sistem akan mengeksekusi Kode 5.29 yang menjelaskan bagaimana server menerima 2 parameter yang diberikan oleh user melalui halaman login yaitu email dan password. Dengan data tersebut dilakukan eksekusi perintah select menggunakan cypher query dengan penjelasan berikut:

Tabel 5.13. Cypher query login member

Query	Penjelasan
MATCH(n:Member {email:{emailParam}, password:{passwordParam}}) RETURN n	Mencari node berlabel Member dengan atribut email dan password yang sesuai dengan inputan user .
RETURN n	Mengembalikan node Member dengan seluruh atributnya

Hasil dari neo4j disimpan dalam variabel *Result*. Untuk mengakses *Result* maka dibuat sebuah array yaitu *memberArr* yang menyimpan nilai dari seluruh keterangan dan atribut node yang dikembalikan dari hasil eksekusi query pada Tabel 5.13 . Nilai atribut *membername* dari node yang dihasilkan disimpan dalam variabel *membername* sebagai input pada proses selanjutnya.

Jika proses login berhasil user akan diarahkan ke halaman login toko untuk melakukan proses selanjutnya dengan menyertakan parameter *membername* pada saat proses login member

5.4.4.4 Pembuatan Halaman Register Toko

Dalam sistem Recommerce, member dapat mendaftarkan lebih dari satu toko (*multi tenant*) dengan token yang berbeda. Oleh karena itu dilakukan implementasi halaman pendaftaran toko dalam website recommerce.

```
<form name="randform" role="form" action="/shop/add?membername=" method="post">

  <div class="dropdown">
    <button class="btn btn-primary dropdown-toggle" type="button" data-toggle="dropdown">E-commerce Platform Ty
    <span class="caret"></span></button>
    <ul class="dropdown-menu">
      <li><a>Woocommerce</a></li>
    </ul>
  </div><br>

  <div class="form-group">
    <label class="sr-only" for="r-form-1-first-name">username</label>
    <input type="text" name="membername" class="r-form-1-first-name form-control" id="membername" value="%{
req.query.membername %}">
  </div>

  <div class="form-group">
    <label class="sr-only" for="r-form-1-first-name">url shop</label>
    <input type="text" name="urlshop" placeholder="url shop" class="r-form-1-first-name form-control" id="urlshop">
  </div>

  <div class="form-group">
    <label class="sr-only" for="r-form-1-about-yourself">Customer Key</label>
    <textarea name="ck" placeholder="Input customer key from your app API."
      class="r-form-1-about-yourself form-control" id="ck"></textarea>
  </div>

  <div class="form-group">
    <label class="sr-only" for="r-form-1-about-yourself">Customer Secret</label>
    <textarea name="cs" placeholder="Input customer secret from your app API."
      class="r-form-1-about-yourself form-control" id="cs"></textarea>
  </div>
```

```

<div class="form-group">
  <label class="sr-only" for="r-form-1-about-yourself">Customer Secret</label>
  <textarea name="cs" placeholder="Input customer secret from your app API."
    class="r-form-1-about-yourself form-control" id="cs"></textarea>
</div>
<div class="form-group">

<button type="button" class="btn" onClick="randomString();" >Generate Token</button>
<label class="sr-only" for="r-form-1-about-yourself">My Token</label>
<input name="randomfield" placeholder=" "
  class="r-form-1-about-yourself form-control" id="randomfield"></input>

  <script language="javascript" type="text/javascript">
function randomString() {
  var chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";
  var string_length = 16;
  var randomstring = '';
  for (var i=0; i<string_length; i++) {
    var rnum = Math.floor(Math.random() * chars.length);
    randomstring += chars.substring(rnum,rnum+1);
  }
  document.randform.randomfield.value = randomstring;
}
</script>
</div>

<button type="submit" class="btn">Create Shop!</button><br><br>

```

Kode 5.30. Implementasi halaman pendaftaran toko

```

1. app.get('/shop', function (req, res){
2.   res.render('shop', {req: req})
3. })

```

Kode 5.31. Rendering halaman pendaftaran toko

```

1. //register the shop
2. app.post('/shop/add',function(req,res){
3.   var membername = req.body.membername;
4.   var urlshop = req.body.urlshop;
5.   var ck = req.body.ck;
6.   var cs = req.body.cs;
7.   var token =req.body.randomfield;
8.   var type = 'woocommerce';
9.   var myshopArr = [];
10.

```

```

11.     session
12. .run('CREATE(n:Shop{urlshop:{urlshopParam}, ck:{
    ckParam}, cs:{csParam}, token:{tokenParam}, type
    :1}) RETURN n',{urlshopParam:urlshop, ckParam:ck
    , csParam:cs, tokenParam:token})
13. .then(function(result){
14.
15. result.records.forEach(function (record){
16. myshopArr.push({
17. token: record._fields[0].properties.token
18.     });
19.     });
20.
21. session
22. .run('MATCH (a:Member),(b:Shop) WHERE a.membername
    ={membernameParam} AND b.token = {tokenParam}
    CREATE (a)-[r:OWN]-
    >(b) RETURN r',{membernameParam:membername, toke
    nParam:token})
23.     .then(function(result){
24.
25.         session.close();
26.     })
27.     .catch(function(err){
28.         console.log(err);
29.     });
30.     res.redirect('/dashboard?token='+myshopArr[0]
    ['token']+'&membername='+membername);
31. })
32. .catch(function(err){
33.     console.log(err);
34. });
35. });

```

Kode 5.32. proses insert toko baru pada graph

5.4.4.5 Pembuatan Halaman Login Toko

```
<div class="row">
  <div class="col-sm-6 col-sm-offset-3 r-form-1-box wow fadeInUp">
    <div class="r-form-1-top">
      <div class="r-form-1-top-left">
        <h3>Login Now / <a href="http://localhost:3000/shop" style="color: #fff" ><b>Create Shop</b> </a></h3>
        <p>Fill in the form below to get recommerce access:</p>
      </div>
      <div class="r-form-1-top-right">
        <i class="fa fa-pencil"></i>
      </div>
    </div>
    <div class="r-form-1-bottom">
      <form name="randform" role="form" action="/member/matchshop" method="post">
        <div class="form-group">
          <label class="sr-only" for="r-form-1-last-name">Token</label>
          <input type="text" name="token" placeholder="Token" class="r-form-1-last-name form-control" id="token">
        </div>
        <button type="submit" class="btn">Login!</button><br><br>
      </form>
    </div>
  </div>
</div>
</div>
</div>
</div>
```

Kode 5.33. Implementasi halaman login toko

```
1. app.get('/loginshop, function (req, res){
2.   res.render(loginshop)
```

Kode 5.34. Rendering halaman login toko

```
1. //login shop
2. app.post('/member/matchshop',function(req,res){
3.   var token = req.body.token;
4.   var shopArr = [];
5.
6.   session
7.     .run('MATCH(n:Shop{token:{tokenParam}})<-
8.     [r:OWN]-(m:Member) RETURN m,n',{
9.     tokenParam:token})
10.    .then(function(result){
11.
12.      result.records.forEach(function (record){
13.
14.        shopArr.push({
15.          token:
16.          record._fields[1].properties.token,
17.          urlshop:
18.          record._fields[1].properties.urlshop,
```

```

18.         totalproduk:
19.         record._fields[1].properties.totalproduk,
20.         totalcust:
21.         record._fields[1].properties.totalcust,
22.         totalorder:
23.         record._fields[1].properties.totalorder,
24.         membername:
25.         record._fields[0].properties.membername
26.     });
27. });
28. res.redirect('/dashboard?token='+shopArr[0]
    ['token']+'&membername='+shopArr[0]['membername']
   +'&url='+shopArr[0]['urlshop']+'&tp='+shopArr[0]
    ['totalproduk']+'&tc='+shopArr[0]['totalcust']
   +'&to='+shopArr[0]['totalorder']);
29.
30. })
31. .catch(function(err){
32.     console.log(err);
33. });
34.
35. });

```

Kode 5.35. proses login toko pada graph

5.4.4.6 Pembuatan Halaman Dashboard

5.4.4.6.1 Halaman Dashboard Utama

```

<div class="sidebar-wrapper">
  <div class="logo">
    <a href="http://www.creative-tim.com" class="simple-text">
      Recommerce
    </a>
  </div>
  <br><p style="text-align:center"><%= req.query.url %></p>
  <ul class="nav">
    <li class="active">
      <a href="dashboard?token=<%= req.query.token %>&membername=<%= req.query.membername %>&url=<%= req.query.url %>&tp=<%= req.query.tp %>&tc=<%= req.query.tc %>&to=<%= req.query.to %>">
        <i class="ti-panel"></i>
        <p>Dashboard</p>
      </a>
    </li>
    <li>
      <a href="user?token=<%= req.query.token %>&membername=<%= req.query.membername %>&url=<%= req.query.url %>&tp=<%= req.query.tp %>&tc=<%= req.query.tc %>&to=<%= req.query.to %>">
        <i class="ti-user"></i>
        <p>User Profile</p>
      </a>
    </li>
    <li>
      <a href="table?token=<%= req.query.token %>&membername=<%= req.query.membername %>&url=<%= req.query.url %>&tp=<%= req.query.tp %>&tc=<%= req.query.tc %>&to=<%= req.query.to %>">
        <i class="ti-view-list-alt"></i>
        <p>Table List</p>
      </a>
    </li>
  </ul>

```

```

        </div>
        <div class="col-xs-7">
            <div class="numbers">
                <p>Customers</p>
                <%= req.query.tc %>
            </div>
        </div>
    </div>
    <div class="footer">
        <hr />
        <div class="stats">
            <i class="ti-calendar"></i> Jumlah Pelanggan Toko
        </div>
    </div>
</div>
<div class="col-lg-4 col-sm-6">
    <div class="card">
        <div class="content">
            <div class="row">
                <div class="col-xs-5">
                    <div class="icon-big icon-danger text-center">
                        <i class="ti-package"></i>
                    </div>
                </div>
                <div class="col-xs-7">
                    <div class="numbers">
                        <p>Products</p>
                        <%= req.query.tp %>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Kode 5.36. Implementasi halaman dashboard utama

```

1. app.get('/dashboard, function (req, res){
2.   res.render(dashboard)

```

Kode 5.37. Rendering halaman dashboard

```

1. session
2.   .run('MATCH(n:Shop{token:{tokenParam}})<-
   [r:OWN]-
   (m:Member) RETURN m,n',{ tokenParam:token})
3.   .then(function(result){
4.
5.     result.records.forEach(function (record){
6.       shopArr.push({

```



```

7.         token: record._fields[1].properties.to
      ken,
8.         urlshop: record._fields[1].properties.
      urlshop,
9.         totalproduk: record._fields[1].propert
      ies.totalproduk,
10.        totalcust: record._fields[1].propertie
      s.totalcust,
11.        totalorder: record._fields[1].properti
      es.totalorder,
12.        membername: record._fields[0].properti
      es.membername
13.    });
14.    });
15.    //console.log(memberArr[0]['cs']);
16.    res.redirect('/dashboard?token='+shopArr[0]
      ['token']+'&membername='+shopArr[0]['membername']
     +'&url='+shopArr[0]['urlshop']+'&tp='+shopArr[0]
      ['totalproduk']+'&tc='+shopArr[0]['totalcust']
     +'&to='+shopArr[0]['totalorder']);

```

Kode 5.38. Mengambil data toko berdasarkan token

```

1. <li>
2.   <a href="dashboard?token=<%= req.query.token %>
      &membername=<%= req.query.membername %>&url=<%=
      req.query.url %>&tp=<%= req.query.tp %>&tc=<%= r
      eq.query.tc %>&to=<%= req.query.to %>">
3.   <i class="ti-view-list-alt"></i>
4.   <p>Table List</p>
5. </a>
6. </li>

```

Kode 5.39. parameter yang diddapatkan dari halaman dahboard

5.4.4.6.2 Halaman Profil Member

```
<div class="col-lg-8 col-md-7">
  <div class="card">
    <div class="header">
      <h4 class="title">Shop's Profile</h4>
    </div>
    <div class="content">
      <form>
        <div class="row">
          <div class="col-md-5">
            <div class="form-group">
              <label>Company</label>
              <input type="text" class="form-control border-input" disabled placeholder="Url Shop"
                value="<%= req.query.url %%">
            </div>
          </div>
          <div class="col-md-3">
            <div class="form-group">
              <label>Username</label>
              <input type="text" class="form-control border-input" placeholder="Membername"
                value="<%= req.query.membername %%">
            </div>
          </div>
          <div class="col-md-4">
            <div class="form-group">
              <label for="exampleInputEmail">Token</label>
              <input type="text" class="form-control border-input" placeholder="<%= req.query.token
                %%">
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>
```

Kode 5.40. Implementasi halaman profil member

```
1. <li class="active">
2. <a href="user?token=<%= req.query.token %>&membe
   rname=<%= req.query.membername %>&url=<%= req.qu
   ery.url %>&tp=<%= req.query.tp %>&tc=<%= req.que
   ry.tc %>&to=<%= req.query.to %>">
3. <i class="ti-user"></i>
4. <p>User Profile</p>
5. </a>
6. </li>
```

Kode 5.41. Paramater yang disimpan pada halaman user

5.4.4.6.3 Halaman Tabel Data

```
1. <li class="active">
2. <a href="table?token=<%= req.query.token %>&memb
   ername=<%= req.query.membername %>&url=<%= req.q
   uery.url %>&tp=<%= req.query.tp %>&tc=<%= req.qu
   ery.tc %>&to=<%= req.query.to %>">
3. <i class="ti-user"></i>
4. <p>User Profile</p>
```

5. ``
6. ``

Kode 5.42. Parameter yang di simpan pada halaman tabel data

Halaman ini sengaja dikosongkan

BAB VI

HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan hasil serta analisis terhadap hasil yang diperoleh dari proses implementasi yang telah dibahas pada bab sebelumnya.

6.1 Hasil

Berikut ini dijabarkan hasil dari implementasi yang telah dilakukan dalam sub bab sebelumnya serta proses pengujian yang dilakukan pada aplikasi. Pengujian yang dilakukan yaitu functional testing dengan metode black box dan pengujian rekomendasi kepada expert.

6.1.1 Pengujian Fungsional (Functional Testing)

Pengujian sistem dilakukan untuk mengetahui apakah fitur yang dibuat pada aplikasi dapat berjalan dengan baik. Semua fitur akan dicoba, tetapi dokumentasi pada buku ini hanya pada fitur-fitur utama aplikasi saja. Tabel berikut daftar fitur yang diuji.

Tabel 6.1. fitur Recommerce

No	Fitur
F01	Register Member
F02	Login Member
F03	Pendaftaran Toko
F04	Login Toko
F05	Mengambil API order
F06	Mengambil API product
F07	Mengambil API pelanggan

F08	Update Data Pelanggan
F09	Update Data Produk
F10	Update Data Order
F11	Mengambil rekomendasi
F12	Lihat Daftar Produk
F13	Lihat Daftar Pelanggan
F14	Lihat Profil Toko
F15	Lihat profil member

Tabel 6.2. Pengujian Fungsional Aplikasi

Process	Input	Expected Result	Actual Result
Manajemen User Akses			
F01. Register Member	Member name, email password	Member name, email dan password tersipan dalam database graf	Member name, email dan password tersipan dalam database graf
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F02. Login Member	Email, password	<ul style="list-style-type: none"> Login berhasil jika email & password sesuai 	<ul style="list-style-type: none"> Login berhasil jika email & password sesuai

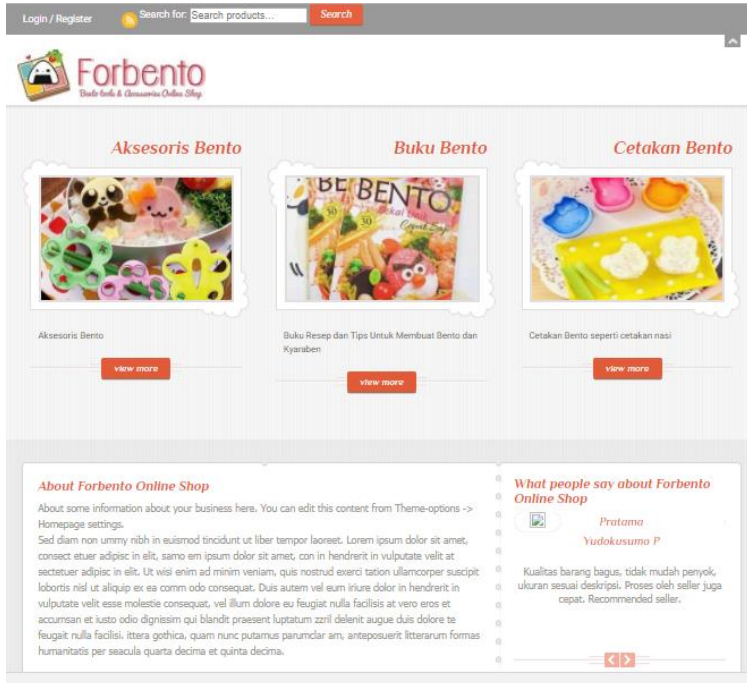
		<ul style="list-style-type: none"> Gagal login jika email & password tidak sesuai 	<ul style="list-style-type: none"> Gagal login jika email & password tidak sesuai
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F03. Mendaftarkan toko	url toko, customer key, customer secret, token	<ul style="list-style-type: none"> url toko, customer key, customer secret, token disimpan dalam database Terbentuk hubungan antara member dan toko 	<ul style="list-style-type: none"> url toko, customer key, customer secret, token disimpan dalam database Terbentuk hubungan antara member dan toko
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F04. Login toko	Token	Berhasil login ke dalam dashboard toko	Berhasil login ke dalam dashboard toko
F05. Mengambil data order	Customer, key, customer secret, url shop, token	Data order dalam format json didapatkan dan disimpan dalam file token_ord.json	Data order dalam format json didapatkan dan disimpan dalam file token_ord.json
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		

F06. Mengambil data pelanggan	Customer, key, customer secret, url shop, token	Data pelanggan dalam format json didapatkan dan disimpan dalam file token_cust.json	Data order dalam format json didapatkan dan disimpan dalam file token_cust.json
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F07. Mengambil data produk	Customer, key, customer secret, url shop, token	Data order dalam format json didapatkan dan disimpan dalam file token_prod.json	Data order dalam format json didapatkan dan disimpan dalam file token_prod.json
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F08. Update Data Produk	File token _prod.json, token	Database graf terupdate. Terdapat node produk baru dari data json	Database graf terupdate. Terdapat node produk baru dari data json
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F09. Update Data Pelanggan	File token _cust.json, token	Database graf terupdate. Terdapat node pelanggan baru dari data json	Database graf terupdate. Terdapat node pelanggan baru dari data json
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		

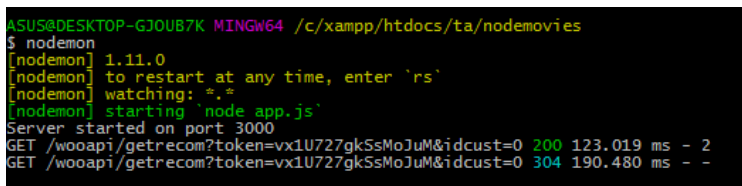
F10. Update Data Order	File token _ord.json, token	Database graf terupdate. Terdapat <i>relationship</i> baru antara data pelanggan dan produk dari data json dalam setiap order	Database graf terupdate. Terdapat <i>relationship</i> baru antara data pelanggan dan produk dari data json dalam setiap order
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F11. Mengambil Rekomendasi	Id cust yang sedang login, token	Sistem memberikan hasil produk yang direkomendasikan	Sistem memberikan hasil produk yang direkomendasikan
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F12. Lihat Daftar Produk	Username	Menampilkan daftar produk setiap toko	Menampilkan daftar produk setiap toko
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F13. Lihat Daftar Pelanggan	Username	Menampilkan daftar pelanggan setiap toko	Menampilkan daftar pelanggan setiap toko
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		

F14. Lihat Profil Toko	Token	Menampilkan detail informasi toko	Menampilkan detail informasi toko
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		
F15. Lihat Profil Member	membername	Menampilkan detail informasi member	Menampilkan detail informasi member
	Sesuai dengan ekspektasi. Setelah dilakukan uji coba tidak ditemukan error.		

Selain keseluruhan fitur pada Tabel 6.2, juga dilakukan pengujian integrasi klien dan server dalam mengambil rekomendasi dari web service. Hasil yang didapatkan adalah klien mengambil 6 rekomendasi produk teratas saat pelanggan toko sedang berada pada halaman utama website klien (website forbento) dan dalam keadaan telah login. Sedangkan apabila pelanggan toko belum login pada website client, halaman bagian rekomendasi akan disembunyikan. Berikut ini merupakan hasil rekomendasi yang disajikan dalam website klien.




Gambar 6.1. Hasil Rekomendasi Produk saat tidak ada user yang login




Gambar 6.2. Tampilan console server saat mengambil rekomendasi

Gambar 6.1 dan Gambar 6.2 menunjukkan hasil rekomendasi saat tidak ada pelanggan yang sedang login, ataupun pelanggan yang belum banyak melakukan transaksi sehingga tidak ada rekomendasi yang didapatkan. Pada tampilan console diketahui bahwa server mengambil informasi dari token toko dan id pelanggan yang sedang login.

Welcome IbuFera | My Account | Logout
Search for: Search products... Search


Forbento
Snack Tools & Accessories Online Shop


Aksesoris Bento



Aksesoris Bento

[view more](#)


Buku Bento



Buku Resep dan Tips Untuk Membuat Bento dan Kyaraben

[view more](#)

Cetakan Bento



Cetakan Bento seperti cetakan nasi

[view more](#)

About Forbento Online Shop

About some information about your business here. You can edit this content from Theme-options -> Homepage settings.


Sed diam non ummy nibh in euismod trincidunt ut liber tempor laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing in elit, sams em ipsum dolor sit amet, con in hendrerit in vulputate velit at sedctetur adipsc in elit. Ut visi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea comm odio consequat. Duis autem vel eum inire dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et justo odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugiat nulla facilisis. Itera gothica, quam nunc putamus panumdar am, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima.

What people say about Forbento Online Shop

Protama Yudokusuma P

Kualitas barang bagus, tidak mudah penyok, ukuran sesuai deskripsi. Proses oleh seller juga cepet. Recommended seller.

Special Product recommendation for you




Bento Cutter - Mold Set Animals Elephant

Klik untuk detail produk

Rp 80000

[Read More](#)




Die Cut Nori Winnie the Pooh by Face KAI

Klik untuk detail produk

Rp 30000

[Read More](#)




Die Cut Nori Mickey Isl 50wajah

Klik untuk detail produk

Rp 30000

[Read More](#)




Catering Box Bento Hiro Small LB115

Klik untuk detail produk

Rp 3000

[Read More](#)




Catering Box Bento Hiro LB-118 (BLACK)

Klik untuk detail produk

Rp 3000

[Read More](#)



Pinset Ujung Bengkok

Klik untuk detail produk

Rp 19000

[Read More](#)

Gambar 6.3. Hasil Rekomendasi produk untuk IbuFera(id cust:880)

6.1.2 Validasi kode

Dalam hal ini dilakukan proses validasi dari kode yang digunakan untuk menghitung kemiripan antar user menggunakan cosine similarity dan euclidian distance. Proses validasi dilakukan dengan membandingkan hasil perhitungan manual dari kedua pelanggan dan nilai kemiripan yang dihasilkan dari kode program yang digunakan.

6.1.2.1 Validasi kode cosine similarity

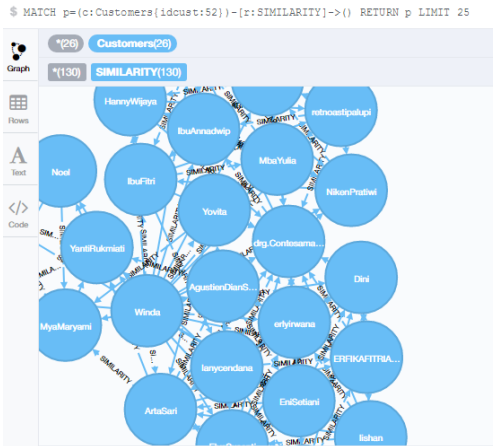
Metode pertama adalah cosine similarity yang dieksekusi melalui program dengan kode berikut:

```

1. function addSimil(token){
2.   session
3.   .run('MATCH (u1:Customers)-[x:BUY]-
      >(p:Product)<-[y:BUY]-(u2:Customers)
4.   WITH SUM(x.rec_order * y.rec_order) AS DotProduct,
5.   SQRT(REDUCE(xDot = 0, i IN COLLECT(x.
      rec_order ) | xDot + toInt(i^2))) AS xLength,
6.   SQRT(REDUCE(yDot = 0, j IN COLLECT(y.
      rec_order ) | yDot + toInt(j^2))) AS yLength,
7.   u1, u2
8.   CREATE UNIQUE (u1)-[s:SIMILARITY]-(u2)
9.   SET s.value = DotProduct / (xLength * yLength)',
      {tokenParam:token})
10.  .then(function(result){
11.    session.close();
12.  })
13.
14.  .catch(function(err){
15.    console.log(err);
16.  });
17. }
```

Kode 6.1. Validasi kode cosine similarity

Dari eksekusi Kode 6.1, didapatkan hasil pembentukan similarity untuk semua pelanggan yang ada di dalam database yaitu dapat dilihat pada Gambar 6.4 di bawah ini.



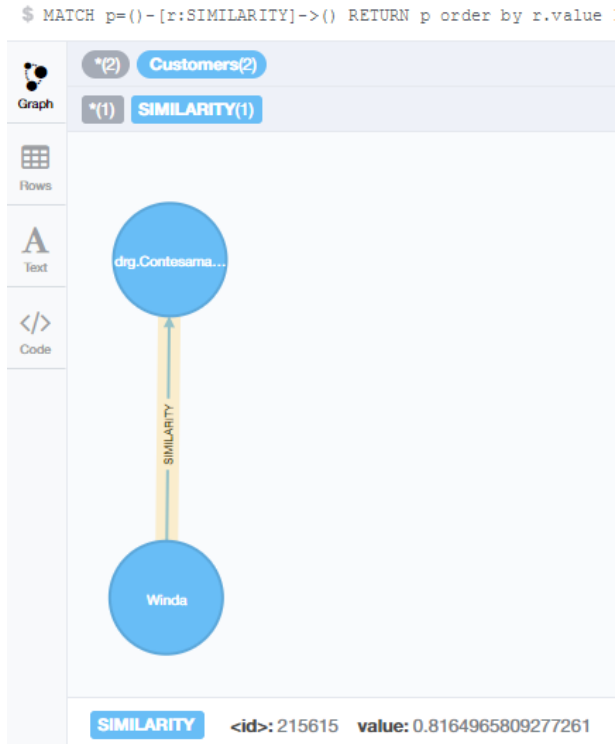
Gambar 6.4. Hasil Pembuatan similarity keseluruhan

Setelah dilakukan pembuatan similarity, maka diambil 2 pengguna secara acak untuk dilakukan perbandingan perhitungan dengan program dan hasil perhitungan manual.

Dengan hasil eksekusi Kode 6.1, dapat dilihat hasil similarity dari dua pelanggan tertentu. Untuk percobaan ini, didapatkan user berikut sebagai sample.

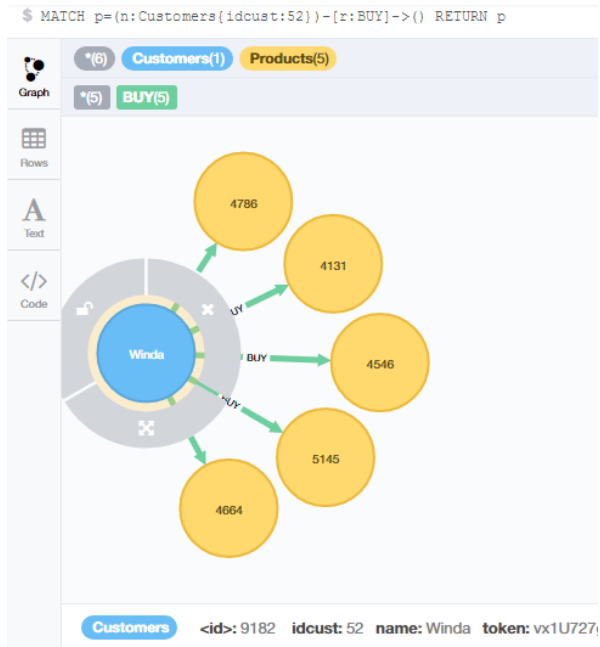
- a. Winda (user id: 52)
- b. drg.Contesamaruhawa (user id: 733)

Uji similarity dengan program menghasilkan nilai **0.8164965809277261** atau 0,82. Hasil visualisasi graf antara kedua user tersebut dipaparkan pada Gambar 6.5



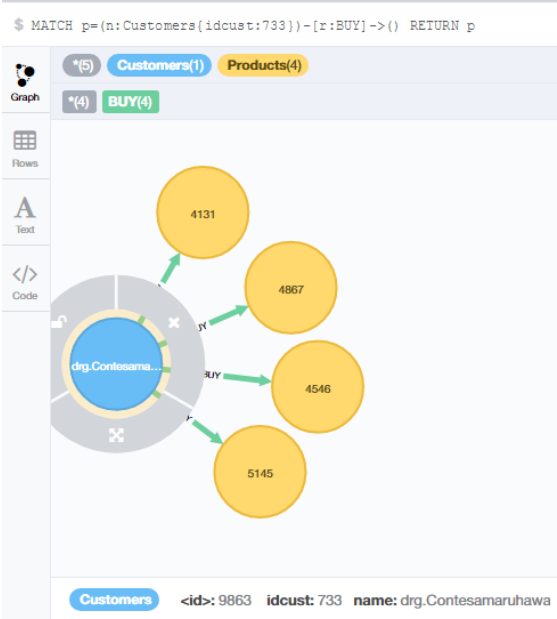
Gambar 6.5. Visualisasi similarity melalui program

Untuk melakukan perbandingan dengan perhitungan manual, maka terlebih dahulu dilakukan pencarian transaksi yang pernah dilakukan oleh masing-masing pelanggan. Gambar 6.6 menunjukkan daftar transaksi yang dilakukan oleh Winda, dengan id 52.



Gambar 6.6. visualisasi graf transaksi pelanggan id 52

Dari hasil tersebut, dicatat produk yang dibeli oleh pelanggan id 52 dan nilai recurrence order dari setiap pembeliannya. Begitu juga dengan hasil transaksi pembelian yang dilakukan oleh pelanggan id 733 yang ditunjukkan pada Gambar 6.7



Gambar 6.7. Visualisasi pelanggan id 733

Setelah itu, dilakukan pencatatan nilai recurrence order dari setiap pembelian pada Tabel 6.3.

Tabel 6.3. Matriks transaksi pembelian dari kedua pelanggan

idcust/idprod	4131	4786	4546	5145	4664	4867
733	1		1	1		1
52	4	1	1	1	4	

Dari data tersebut,hanya diambil produk yang dibeli oleh kedua pelanggan untuk kemudian dihitung kemiripan melalui metode cosine similarity.

Tabel 6.4. Matriks transaksi produk yang sama

Idcust/idprod	4131	4546	5145
733	1	1	1
52	4	1	1

Hasil perhitungan kemiripan dengan cosine similarity dapat dilihat pada penjabaran di bawah ini.

$$\text{Similarity}(x, y) = \frac{\sum_{i=1}^n A1xB1}{\sqrt{\sum_{i=1}^n A1^2} \times \sqrt{\sum_{i=1}^n B1^2}}$$

$$\text{Similarity}(x, y) = \frac{\sum_{i=1}^n A1xB1}{\sqrt{\sum_{i=1}^n A1^2} \times \sqrt{\sum_{i=1}^n B1^2}}$$

$$\text{Similarity}(733, 52) = \frac{(1 \times 4) + (1 \times 1) + (1 \times 1)}{\sqrt{1^2 + 1^2 + 1^2} \times \sqrt{4^2 + 1^2 + 1^2}}$$

$$\text{Similarity}(733, 52) = \frac{6}{\sqrt{3} \times \sqrt{18}}$$

$$\text{Similarity}(733, 52) = \mathbf{0,816496}$$

Berdasarkan perhitungan yang dilakukan baik melalui program ataupun perhitungan manual, dihasilkan hasil similarity yang sama yaitu **0,82**

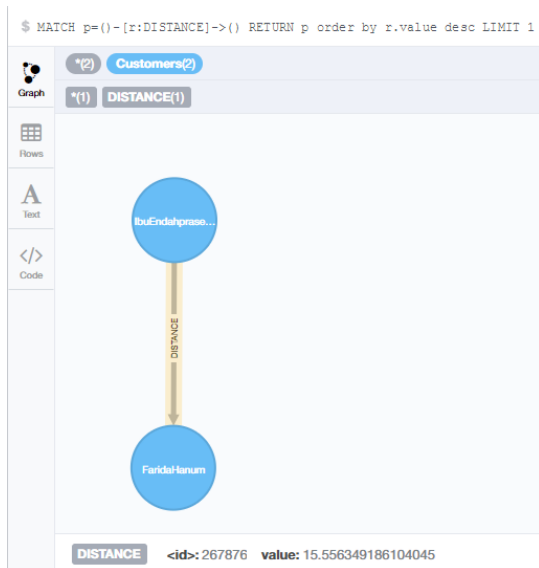
6.1.2.2 Validasi Kode euclidian distance

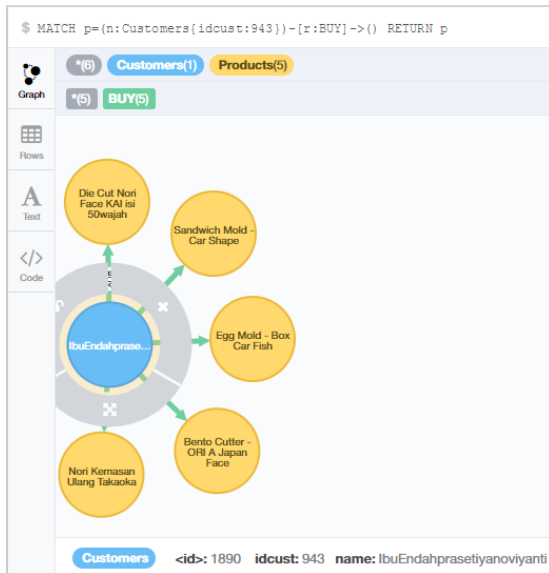
```

1. function addSimil(token){
2. session
3. .run('MATCH (u1:Customers)-[x:BUY]-
   >(p:Product)<-[y:BUY]-(u2:Customers)
4. WHERE id(u1)<id(u2)
5. WITH SQRT(SUM((x.rec_order - y.rec_order)^2))
   AS euc, u1, u2
6.
7. CREATE UNIQUE (u1)-[d:DISTANCE]-(u2)
8. SET d.value = euc', {tokenParam:token})
9. .then(function(result){
10.     session.close();
11. })
12.
13. .catch(function(err){
14.     console.log(err);

```

```
15.   });  
16. }  
17.
```





idcust/idprod	5131	4668	4786	5303	5145
943	1	1	1	1	1
303		12	12		

idcust/idprod	4668	4786
943	1	1
303	12	12

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

$$d(303, 943) = \sqrt{(12 - 1)^2 + (12 - 1)^2}$$

$$d(303, 943) = \sqrt{242}$$

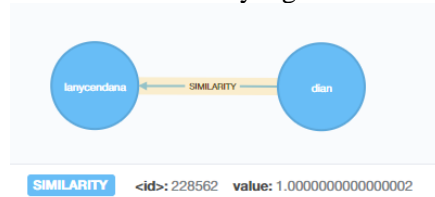
$$d(303, 943) = \mathbf{15,556349186}$$

6.1.3 Validasi Kemiripan pelanggan

Proses ini bertujuan untuk membuktikan perbedaan hasil rekomendasi dari derajat kemiripan pelanggan yang berbeda.

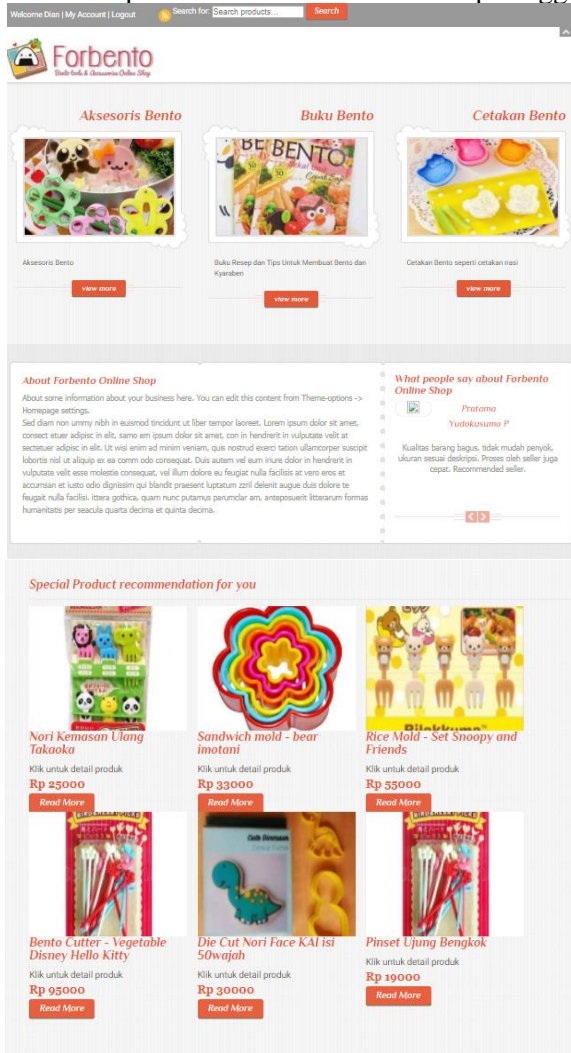
6.1.3.1 Pembuktian pelanggan yang mirip

Dalam proses ini dibuktikan bahwa hasil rekomendasi antara dua user yang memiliki kemiripan yang tinggi cenderung mendapatkan hasil rekomendasi yang sama



Gambar 6.8. Sampel kemiripan pelanggan tertinggi


Berikut ini merupakan hasil rekomendasi dari pelanggan dian





Gambar 6.9. Hasil rekomendasi untuk pelanggan dian


Berikut ini merupakan hasil rekomendasi dari pelanggan
lanycendana

Welcome Lanycendana | My Account | Logout
Search for: Search products... Search



Forbento
Bento Tools & Accessories Online Shop

Aksesoris Bento

Aksesoris Bento
[View more](#)


Buku Bento

Buku Resep dan Tips Untuk Membuat Bento dan Kyanaben
[View more](#)


Cetakan Bento

Cetakan Bento seperti cetakan nasi
[View more](#)


About Forbento Online Shop
About some information about your business here. You can edit this content from Theme-options -> Homepage settings.
Sed diam non ummy nibh in euismod tristique ut liber tempor laoreet. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.


What people say about Forbento Online Shop

Pratomo
Yusokusumo P
Kualitas barang bagus, tidak mudah penyok, ukuran sesuai deskripsi. Proses oleh seller juga cepat. Recommended seller.


Special Product recommendation for you



Nori Kemasn Ulang Takaoka
Klik untuk detail produk
Rp 25000
[Read More](#)


Sandwich mold - bear imotani
Klik untuk detail produk
Rp 33000
[Read More](#)


Rice Mold - Set Snoopy and Friends
Klik untuk detail produk
Rp 55000
[Read More](#)


Bento Cutter - ORI A Japan Face
Klik untuk detail produk
Rp 200000
[Read More](#)


Rice Mold Cute Cat with Nori Cutter
Klik untuk detail produk
Rp 165000
[Read More](#)


Bento Cutter - Vegetable Disney Hello Kitty
Klik untuk detail produk
Rp 95000
[Read More](#)

Gambar 6.10. Hasil Rekomendasi pelanggan Lanycendana

Dari Gambar 6.9 dan Gambar 6.10, didapatkan beberapa hasil rekomendasi dengan tidak memperhatikan gambar produk yang dimuat secara acak.

Pembuktian:

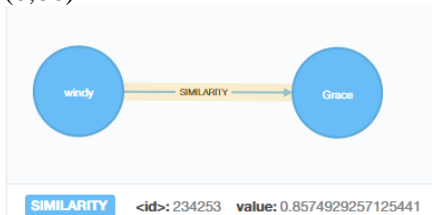
Tabel 6.5. Tabel kemiripan hasil rekomendasi

No	Hasil rekomendasi pelanggan dian	Hasil rekomendasi pelanggan lanycendana	Keterangan
1	Nori kemasan ulang takaoka	Nori kemasan ulang takaoka	Sama
2	Sandwich mold-bear imotani	Sandwich mold-bear imotani	Sama
3	Sandwich mold- Set snoppy and friends	Sandwich mold- Set snoppy and friends	Sama
4	Bento Cutter – Vegetable disney hello kitty	Bento Cutter – Vegetable disney hello kitty	Sama
5	Die Cut Nori Face KAI isi 50 wajah	Bento Cutter- ORI A Japan Face	Berbeda
6	Pinset ujung bengkok	Rice Mold Cute Cat with Nori Cutter	Berbeda

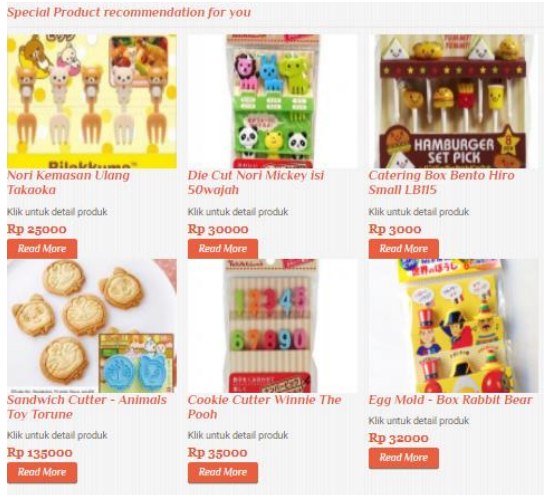
Dari Tabel 6.5 dapat diketahui bahwa hasil rekomendasi dari kedua user dengan kemiripan yang tertinggi memiliki hasil rekomendasi yang sangat mirip (mencapai nilai 1) yaitu 4 produk rekomendasi yang sama dan 2 produk berbeda

6.1.3.2 Pembuktian pelanggan yang tidak mirip

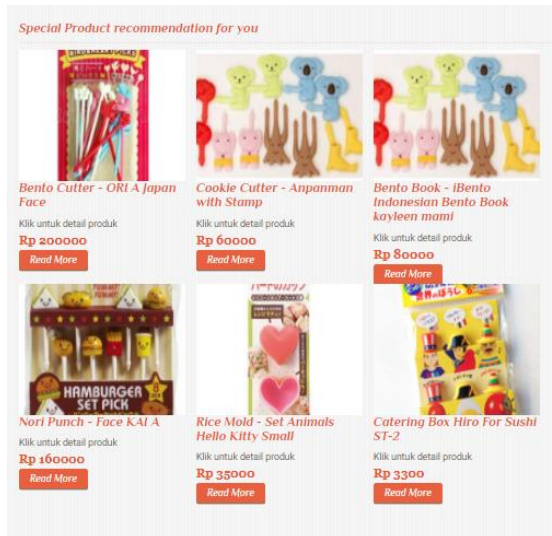
Dalam tahap ini, akan dibuktikan hasil rekomendasi dari dua pelanggan yang tidak mirip atau memiliki derajat kemiripan yang rendah (0,86)



Gambar 6.11. Visualisasi 2 pelanggan yang tidak mirip



Gambar 6.12. Hasil reomendasi pelanggan windy



Gambar 6.13. Hasil rekomendasi pelanggan grace

Dari kedua gambar diatas dapat dicatat perbandingan hasil rekomendasi berikut:

No	Hasil rekomendasi pelanggan windy	Hasil rekomendasi pelanggan grace	Keterangan
1	Nori Kemasan Ulang Takaoka	Bento Cutter – ORI Face	Berbeda
2	Die Cut Nori Mickey isi 50 wajah	Anpanman with Stamp	Berbeda
3	Catering Box Bento Hiro Small LB115	Bento Book -iBento Indonesian Bento Book	Berbeda
4	Sandwich Cutter – Animals Toy Torune	Nori Punch -Face KAI A	Berbeda
5	Cookie Cutter Winnie The Pooh	Rice Mold- Set Animals Hello Kitty Small	Berbeda
6	Egg Mold – Box Rabbit Bear	Catering Box Hiro for Sushi ST-2	Berbeda

6.1.4 Pengujian Rekomendasi

Pengujian Rekomendasi Pengujian rekomendasi dilakukan untuk mengetahui seberapa tepat rekomendasi yang dihasilkan terhadap preferensi pelanggan. Pengujian dilakukan oleh expert yang memiliki pengalaman dan pengetahuan tentang pola perilaku setiap pelanggan dalam melakukan transaksi pada Toko Online Forbento yaitu Ibu Rahayu Fatnawati sebagai pemilik Toko Online Forbento

Adapun pengujian dilakukan dengan membandingkan hasil rekomendasi dari algoritma customer based collaborative filtering. Pengujian juga bertujuan membandingkan jenis perhitungan yang paling efektif dan pengaruh faktor tertentu yang. Sehingga dalam proses pengujian ini akan ditetapkan 3 buah objective yaitu sebagai berikut:

1. Mengetahui waktu rata-rata yang dibutuhkan untuk menghasilkan rekomendasi.
2. Mengetahui ketepatan rekomendasi yang dihasilkan dari algoritma collaborative filtering terhadap pelanggan Forbento

3. Membandingkan penggunaan perhitungan kesamaan pelanggan yang berbeda antara cosine similarity dan euclidian distance untuk menghasilkan rekomendasi yang lebih baik
4. Mengetahui faktor faktor yang mempengaruhi ketepatan rekomendasi dalam studi kasus Forbento

6.1.4.1 Waktu pemrosesan rekomendasi

Setiap rekomendasi yang diambil dari berbagai pelanggan yang berbeda memiliki perbedaan waktu yang berbeda pula. Setiap pelanggan memiliki kompleksitas graf yang berbeda sehingga akan berdampak pada waktu pemrosesan rekomendasi. Oleh karena itu jumlah relasi merupakan salah satu faktor yang mempengaruhi jangka waktu pengambilan rekomendasi. Pada penelitian ini dilakukan pengambilan rekomendasi dari 50 id yang berbeda dengan pemilihan berdasarkan tingkat kompleksitasnya yang digambarkan dengan jumlah node dan relasi yang terhubung dari setiap id pelanggan

No	Id pelanggan	Jumlah relasi - similarity	Jumlah relasi - buy	time neo4j (ms)	time node js (ms)	Keterangan
1	52	104	5	9	187,4	Relasi terbanyak dalam database graf
2	886	0	0	4	158,7	Tidak memiliki relasi
3	880	74	5	6	137,9	Sampel acak
4	18	81	4	8	154,7	Sampel acak
5	952	24	5	8	144,1	Sampel acak
6	888	37	1	7	155	Sampel acak
7	874	23	1	5	142,8	Sampel acak
8	878	20	3	5	146,4	Sampel acak
9	908	14	1	4	125,2	Sampel acak

No	Id pelanggan	Jumlah relasi - similarity	Jumlah relasi - buy	time neo4j (ms)	time node js (ms)	Keterangan
10	885	56	5	8	147,9	Sampel acak
11	470	23	1	18	138	Sampel acak
12	118	0	0	4	110	Sampel acak
13	75	26	2	3	123,1	Sampel acak
14	532	23	1	40	132,7	Sampel acak
15	600	23	2	22	134,5	Sampel acak
16	85	26	2	38	142,3	Sampel acak
17	247	28	1	2	122,5	Sampel acak
18	599	29	2	107	132,2	Sampel acak
19	230	56	2	61	117,4	Sampel acak
20	861	22	4	15	125	Sampel acak
21	147	72	6	43	114,7	Sampel acak
22	169	77	3	18	166,7	Sampel acak
23	626	21	1	5	108,1	Sampel acak
24	873	17	1	9	137,2	Sampel acak
25	542	59	3	52	124,3	Sampel acak
26	471	84	5	8	150,5	Sampel acak
27	267	25	2	4	123	Sampel acak
28	188	32	3	4	105,3	Sampel acak
29	735	20	1	3	118,2	Sampel acak
30	41	20	1	3	142	Sampel acak
31	871	10	1	4	122	Sampel acak
32	45	65	3	5	103	Sampel acak
33	776	91	4	53	118	Sampel acak
34	276	45	2	35	118	Sampel acak

No	Id pelanggan	Jumlah relasi - similarity	Jumlah relasi - buy	time neo4j (ms)	time node js (ms)	Keterangan
35	723	23	2	102	133	Sampel acak
36	621	53	2	83	111	Sampel acak
37	424	23	2	8	101	Sampel acak
38	877	20	1	13	136	Sampel acak
39	801	20	1	12	129	Sampel acak
40	943	97	5	31	138	Sampel acak
41	778	58	4	9	107	Sampel acak
42	487	51	3	9	136	Sampel acak
43	422	25	1	6	123	Sampel acak
44	216	21	1	5	102	Sampel acak
45	498	8	1	3	176	Sampel acak
46	319	25	1	5	103	Sampel acak
47	494	24	1	3	100	Sampel acak
48	560	62	5	7	157	Sampel acak
49	409	50	3	10	112	Sampel acak
50	721	3	1	4	121	Sampel acak
Rata-rata		53,5	3	6,5	154,2	

Dari beberapa percobaan tersebut didapatkan waktu eksekusi rata-rata dari neo4j adalah 6,5 ms, sedangkan waktu pemrosesan untuk mendapatkan array dari node js adalah 154,2 ms.

6.1.4.2 Pemilihan Subjek Pengujian

Pemilihan pelanggan yang akan diuji dilakukan dengan beberapa seleksi untuk menjamin pemerataan hasil dan

mengurangi subjektifitas expert. Untuk menentukan kriteria pemilihan subjek pengujian, dilakukan wawancara untuk mengetahui pengaruh jenis pelanggan terhadap preferensinya/hasil rekomendasi.

Berdasarkan hasil wawancara, didapatkan bahwa pelanggan Toko Online Forbento dapat dikelompokkan menjadi 2 jenis tipe pelanggan yaitu tipe pelanggan catering dan personal.

Adapun segmentasi pelanggan tersebut merupakan salah satu acuan untuk mengetahui preferensi produk yang akan dibeli oleh pelanggan forbento yang membentuk pola perilaku yang dapat diketahui setiap orang. Dari hasil wawancara dapat diketahui bahwa pelanggan dengan tipe catering akan cenderung konsisten membeli produk dengan jenis seperti Rice Mold, Catering Box, Nori punch, Die Cut, Nori, hingga pinset, sedangkan pelanggan tipe personal cenderung memilih produk yang digunakan sendiri seperti sandich mold, bento book, Bento Cutter dan lain sebagainya.

Dengan informasi tersebut, maka dilakukan pemilihan pelanggan secara acak dengan kriteria berikut:

1. Merupakan pelanggan yang loyal, sehingga dikenal oleh expert sebagai pelanggannya serta dapat lebih mudah diketahui preferensinya terhadap rekomendasi. Hal ini ditunjukkan dengan adanya pembelian dengan minimum 4 kali order di forbento
2. Komposisi tipe pelanggan catering dan personal seimbang

Dengan indikator tersebut, maka didapatkan 4 subjek penelitian yang menggunakan pelanggan forbento berikut ini:

Tabel 6.6. Data dan riwayat pembelian pelanggan id 880

Nama Pelanggan : Ibu Fera
ID Pelanggan : 880
History Pembelian
1. Rice Mold - Disney Ska Cars Tomica – Rp. 97.000
2. Rice Mold - Disney Ska Winnie the Pooh – Rp. 97.000
3. Catering Box Bento Hiro Medium Sekat3 DB10B – Rp. 3500
4. Catering Box Hiro For Sushi ST-2 – Rp. 3300

5. Nori Kemasan Ulang Takaoka – Rp. 25.000
Tipe Pelanggan : Catering

Tabel 6.7. Data dan riwayat pembelian pelanggan id 52

Nama Pelanggan : Ibu Winda ID Pelanggan : 52
History Pembelian <ol style="list-style-type: none"> 1. Rice Mold - Set Animals Hello Kitty Small – 35.000 2. Sandwich mold - bear imotani – 33.000 3. Egg Mold - Box Car Fish – 32.000 4. Rice Mold - Set Snoopy and Friends – 55.000 5. Nori Kemasan Ulang Takaoka – 25.000
Tipe Pelanggan : Personal Use

Tabel 6.8. Data dan riwayat pembelian pelanggan id 18

Nama Pelanggan : Ibu HannyWijaya ID Pelanggan : 18
History Pembelian <ol style="list-style-type: none"> 1. Pinset Ujung Bengkok – 19.000 2. Catering Box Bento Hiro Medium Sekat3 DB10B – 3800 (15 item) 3. Bento Cutter - Arrange Set Torune – 120000 4. Nori Kemasan Ulang Takaoka – 25.000
Tipe Pelanggan : Catering

Tabel 6.9. Data dan riwayat pembelian pelanggan id 952

Nama Pelanggan : Ibu shintadwijayanti ID Pelanggan : 952
History Pembelian <ol style="list-style-type: none"> 1. Bento Cutter - ORI A Japan Face - 200000 2. Anpanman with Stamp - 60000 3. Cookie Cutter My Melody – 35000 4. Bento Book - iBento Indonesian Bento Book kayleen mami – 80000 5. Rice Mold - Set Snoopy and Friends – 55000
Tipe Pelanggan : Personal Use

6.1.4.3 Skenario Pengujian

Skenario pengujian dilakukan dengan menyiapkan 16 skenario rekomendasi dari 4 pelanggan dengan karakteristik yang berbeda.

Skenario 1

Expert melakukan penilaian kesesuaian produk hasil rekomendasi dari algoritma collaborative filtering dengan perhitungan kesamaan user menggunakan cosine similarity tanpa menghiraukan nilai recurrence order

Skenario ini diulang dan dilakukan sebanyak 4 kali sesuai jumlah pelanggan yang diuji (id pelanggan: 880, 52, 18, 952).

Untuk menghasilkan rekomendasi maka dijalankan kode query berikut ini:

```
1. MATCH(u2:Customers)-[r:BUY]-
  >(p:Products),(u2:Customers)-[s:SIMILARITY]-
  (u1:Customers {idcust: id_pelanggan })
2. WHERE NOT((u1)-[:BUY]->(p))
3. WITH u1 ,p, u2, s.value AS similarity
4. ORDER BY similarity DESC
5. return u2, u1,p LIMIT 7
```

Kode 6.2. Query cypher rekomendasi skenario 1

Skenario 2

Expert melakukan penilaian kesesuaian produk hasil rekomendasi dari algoritma collaborative filtering dengan perhitungan kesamaan user menggunakan cosine similarity tanpa menghiraukan nilai recurrence order

Skenario ini diulang dan dilakukan sebanyak 4 kali sesuai jumlah pelanggan yang diuji (id pelanggan: 880, 52, 18, 952).

Untuk menghasilkan rekomendasi maka dijalankan kode query berikut ini:


```

1. MATCH (u2:Customers)-[r:BUY]-
   >(p:Products), (u2:Customers)-[s:SIMILARITY]-
   (u1:Customers {idcust:id_pelanggan })
2. WHERE NOT((u1)-[:BUY]->(p))and r.ordervalue>1
3. WITH u1,p, u2,r, s.value AS similarity
4. ORDER BY similarity DESC
5. return distinct p LIMIT 7

```

Kode 6.3. Query cypher rekomendasi skenario 2

Skenario 3

Expert melakukan penilaian kesesuaian produk hasil rekomendasi dari algoritma collaborative filtering dengan perhitungan kesamaan user menggunakan euclidian distance tanpa menghiraukan nilai recurrence order.

Skenario ini diulang dan dilakukan sebanyak 4 kali sesuai jumlah pelanggan yang diuji (id pelanggan: 880, 52,18, 952).

Untuk menghasilkan rekomendasi maka dijalankan kode query berikut ini:

```

1. MATCH (u2:Customers)-[r:BUY]-
   >(p:Products), (u2:Customers)-[s:SIMILARITY]-
   (u1:Customers {idcust:id_pelanggan})
2. WHERE NOT((u1)-[:BUY]->(p))
3. WITH u1,p, u2,r, s.value AS similarity
4. ORDER BY similarity ASC
5. return distinct p LIMIT 7

```

Kode 6.4. Query cypher rekomendasi skenario 3

Skenario 4

Expert melakukan penilaian kesesuaian produk hasil rekomendasi dari algoritma collaborative filtering dengan perhitungan kesamaan user menggunakan euclidian distance dengan menyertakan nilai recurrence order tertinggi

Skenario ini diulang dan dilakukan sebanyak 4 kali sesuai jumlah pelanggan yang diuji (id pelanggan: 880, 52,18, 952).

Untuk menghasilkan rekomendasi maka dijalankan kode query berikut ini:

```
1. MATCH (u2:Customers)-[r:BUY]-
    >(p:Products), (u2:Customers)-[s:SIMILARITY]-
    (u1:Customers {idcust:id_pelanggan})
2. WHERE NOT((u1)-[:BUY]->(p))
3. WITH u1,p, u2,r, s.value AS similarity
4. ORDER BY similarity DESC
5. return distinct p LIMIT 7
```

Kode 6.5. Query cypher rekomendasi skenario 4

6.1.4.4 Hasil Pengujian Rekomendasi Expert

Berikut ini merupakan hasil pengujian rekomendasi untuk setiap test case dan keterangannya

6.1.4.4.1 Test Case 1

Test case 1 dilakukan pengujian dengan skenario 1 untuk pelanggan dengan id 880

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi pertama berdasarkan data riwayat pembelian sebelumnya dan penilaian berdasarkan tipe pelanggan catering. Expert menilai bahwa produk no 2, 3, dan 4 memiliki penilaian yang rendah bagi customer dengan tipe catering seperti Ibu Fera.

Tabel 6.10. Pengujian set rekomendasi test case 1

cosine, without recurrence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Die Cut Nori Face KAI isi 50wajah	30.000	✓
2	Sandwich mold - bear imotani	33.000	✗
3	Sandwich mold - Persegi Tutup	33.000	✗
4	Sandwich mold - Persegi Tutup	33.000	✗
5	Nori Punch - Face KAI A	160.000	✓
6	Catering Box Bento Hiro Medium Tanpa Sekat DB10A	3.500	✓
7	Sandwich Cutter - Hello Kitty Sanrio	40.000	✓

Dalam test case ini, ditemukan 3 produk yang tidak sesuai dan 4 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 1 adalah 57% yang dapat dikategorikan cukup baik

Tabel 6.11. Ringkasan hasil pengujian test case 1

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	3
Jumlah Rekomendasi yang benar	4
Persentase ketepatan rekomendasi	57%

6.1.4.4.2 Test Case 2

Test case 2 dilakukan pengujian dengan skenario 2 untuk pelanggan dengan id 880.

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi kedua berdasarkan data riwayat pembelian sebelumnya dan penilaian berdasarkan tipe pelanggan catering. Expert menilai bahwa produk no 2 memiliki kemungkinan yang rendah untuk dibeli bagi customer dengan tipe catering seperti Ibu Fera.

Tabel 6.12. Hasil uji set rekomendasi test case 2

cosine, with recurrence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Bento Cutter - ORI A Japan Face	200.000	✓
2	Sandwich Cutter - Animals Toy Torune	135.000	✗
3	Rice Mold Cute Cat with Nori Cutter	165.000	✓
4	Rice Mold – Segitiga	35.000	✓
5	Rice Mold – GoPanda	150.000	✓
6	Die Cut Nori - Hello Kitty	30.000	✓
7	Catering Box Bento Hiro Small LB115	3.000	✓

Dalam test case ini, ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 2 adalah 86% yang dapat dikategorikan baik.

Tabel 6.13. Ringkasan hasil pengujian test case 2

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.3 Test Case 3

Test case 3 dilakukan pengujian dengan skenario 3 untuk pelanggan dengan id 880

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi ketiga berdasarkan data riwayat pembelian sebelumnya dan penilaian berdasarkan tipe pelanggan catering. Expert menilai bahwa semua produk memiliki kecenderungan yang tinggi untuk dibeli oleh customer dengan tipe catering seperti Ibu Fera.

Tabel 6.14. Hasil uji set rekomendasi test case 3

euclidian without concurrence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Rice Mold - Disney Ska Lilo Stitch	97.000	✓
2	Cookie Cutter - Lilo and Stitch	35.000	✓
3	Rice Mold - Set Animals Hello Kitty	35.000	✓
4	Cookie Cutter - Doraemon Dorami	35.000	✓
5	Rice Mold - Disney Ska Mickey Mouse	97.000	✓
6	Cookie Cutter Marie Cat	35.000	✓
7	Rice Mold - Set Snoopy and Friends	55.000	✓

Dalam test case ini, tidak ditemukan 1 produkpun yang tidak sesuai sehingga persentase ketepatan rekomendasi dari test case 3 adalah 100% yang dapat dikategorikann sangat baik.

Tabel 6.15. Ringkasan hasil pengujian test case 3

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	0
Jumlah Rekomendasi yang benar	7
Persentase ketepatan rekomendasi	100%

6.1.4.4.4 Test Case 4

Test case 4 dilakukan pengujian dengan skenario 4 untuk pelanggan dengan id 880

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi keempat berdasarkan data riwayat pembelian sebelumnya dan penilaian berdasarkan tipe pelanggan catering. Expert menilai bahwa produk no 2 dan 3 memiliki kemungkinan yang rendah untuk dibeli bagi customer dengan tipe catering seperti Ibu Fera.

Tabel 6.16. Hasil uji set rekomendasi test case 4

euclidian with concurrence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Die Cut Nori Face KAI isi 50wajah	30.000	✓
2	Sandwich mold - bear imotani	33.000	✗
3	Sandwich mold - Persegi Tutup	33.000	✗
4	Rice Mold - Cetakan Nasi Curry Hello Kitty	160.000	✓
5	Rice Mold - Cetakan Nasi Curry Rilakuma	160.000	✓
6	Rice Mold - Cetakan Nasi Curry Snoopy	160.000	✓
7	Die Cut Nori Winnie the Pooh by Face KAI	30.000	✓

Dalam test case ini, ditemukan 2 produk yang tidak sesuai dan 5 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 4 adalah 71% yang dapat dikategorikan baik.

Tabel 6.17. Ringkasan hasil pengujian test case 4

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	2
Jumlah Rekomendasi yang benar	5
Persentase ketepatan rekomendasi	71%

6.1.4.4.5 Test Case 5

Test case 5 dilakukan pengujian dengan skenario 1 untuk pelanggan dengan id 52

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi pertama berdasarkan data riwayat pembelian sebelumnya dan penilaian berdasarkan tipe pelanggan *personal use*. Expert menilai bahwa semua produk memiliki kecenderungan yang tinggi untuk dibeli oleh customer dengan tipe *personal use* seperti Ibu Winda.

Tabel 6.18. Hasil uji set rekomendasi test case 5

cosine, without concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Pinset Ujung Bengkok	19.000	✓
2	Bento Book - Bento Karakter Unik (BKU)	60.000	✓
3	Sandwich Mold - Car Shape	33.000	✓
4	Cookie Cutter - Mickey Minnie Mouse	35.000	✓
5	Egg Mold - Box Rabbit Bear	32.000	✓
6	Bento Cutter - ORI A Japan Face	200.000	✓
7	Rice Mold - Disney Ska Cars Tomica	97.000	✓

Dalam test case ini, tidak ditemukan 1 produkpun yang tidak sesuai sehingga persentase ketepatan rekomendasi dari test case 5 adalah 100% yang dapat dikategorikann sangat baik.

Tabel 6.19. Ringkasan hasil pengujian test case 5

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	0
Jumlah Rekomendasi yang benar	7
Persentase ketepatan rekomendasi	100%

6.1.4.4.6 Test Case 6

Test case 6 dilakukan pengujian dengan skenario 2 untuk pelanggan dengan id 52

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi kedua berdasarkan data riwayat pembelian Ibu Winda sebelumnya dan penilaian berdasarkan tipe pelanggan *personal use*. Expert menilai bahwa produk no 5 memiliki kemungkinan yang rendah untuk dibeli bagi customer dengan tipe *personal use* seperti Ibu Winda.

Tabel 6.20. Hasil uji set rekomendasi test case 6

cosine, with concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Bento Cutter - ORI A Japan Face	200.000	✓
2	Sandwich Cutter - Animals Toy Torune	135.000	✓
3	Bento Cutter - Mold Set Animals Elephant	80.000	✓
4	Sandwich cutter - Animals friend cutter kit	175.000	✓
5	Rice Mold Cute Cat with Nori Cutter – GoPanda	165.000	✗
6	Rice Mold - Segitiga	35.000	✓
7	Bento Cutter - Vegetable Disney Hello Kitty	95.000	✓

Dalam test case ini, hanya ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 6 adalah 86% yang dapat dikategorikan baik.

Tabel 6.21. Ringkasan hasil pengujian test case 6

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.7 Test Case 7

Test case 7 dilakukan pengujian dengan skenario 3 untuk pelanggan dengan id 52

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi ketiga berdasarkan data riwayat pembelian Ibu Winda sebelumnya dan penilaian berdasarkan tipe pelanggan *personal use*. Expert menilai bahwa produk no 2 memiliki kemungkinan yang rendah untuk dibeli bagi customer dengan tipe *personal use* seperti Ibu Winda.

Tabel 6.22. Hasil uji set rekomendasi test case 7

euclidian, without concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Sandwich Mold - Car Shape	33.000	✓
2	Catering Box Bento Hiro Medium Sekat3 DB10B	3.800	✗
3	Rice Mold - Disney Ska Lilo Stitch	97.000	✓
4	Cookie Cutter - Lilo and Stitch	35.000	✓
5	Cookie Cutter - Doraemon Dorami	35.000	✓
6	Rice Mold - Disney Ska Winnie the Pooh	97.000	✓
7	Rice Mold - Disney Ska Mickey Mouse	97.000	✓

Dalam test case ini, hanya ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 7 adalah 86% yang dapat dikategorikan baik.

Tabel 6.23. Ringkasan hasil pengujian test case 7

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.8 Test Case 8

Test case 8 dilakukan pengujian dengan skenario 4 untuk pelanggan dengan id 52

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi keempat berdasarkan data riwayat pembelian Ibu Winda sebelumnya dan penilaian berdasarkan tipe pelanggan *personal use*. Expert menilai bahwa produk no 6 memiliki kemungkinan yang rendah untuk dibeli bagi customer dengan tipe *personal use* seperti Ibu Winda.

Tabel 6.24. Hasil uji set rekomendasi test case 8

euclidian, with concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Bento Book - iBento Indonesian Bento Book kayleen mami	80.000	✓
2	Nori Punch - Lotus Flower	25.000	✓
3	Rice Mold - Disney Ska Winnie the Pooh	97.000	✓
4	Sandwich cutter - Animals friend cutter kit	175.000	✓
5	Cookie Cutter - Hello Kitty 4shape with stamp	60.000	✓
6	Catering Box Bento Hiro Medium Sekat3 DB10B	3.800	✗
7	Nori Punch - Face KAI A	160.000	✓

Dalam test case ini, hanya ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 8 adalah 86% yang dapat dikategorikan baik.

Tabel 6.25. Ringkasan hasil pengujian test case 8

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.9 Test Case 9

Test case 9 dilakukan pengujian dengan skenario 1 untuk pelanggan dengan id 18

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi pertama berdasarkan data riwayat pembelian Ibu HannyWijaya sebelumnya dan penilaian berdasarkan tipe pelanggan *catering*. Expert menilai bahwa produk no 7 memiliki kemungkinan yang rendah untuk dibeli bagi customer dengan tipe *catering* seperti Ibu HannyWijaya.

Tabel 6.26. Hasil uji set rekomendasi test case 9

cosine, without concurrence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Sandwich Mold - Car Shape	33.000	✓
2	Die Cut Nori Mickey isi 50wajah	30.000	✓
3	Rice Mold - Disney Ska Mickey Mouse	97.000	✓
4	Die Cut Nori Face KAI isi 50wajah	30.000	✓
5	Die Cut Nori - Hello Kitty	30.000	✓
6	Rice Mold - Tor Four Animals	80.000	✓
7	Lunch Pick Eyess	60.000	✗

Dalam test case ini, hanya ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 9 adalah 86% yang dapat dikategorikan baik.

Tabel 6.27. Ringkasan hasil pengujian test case 9

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.10 Test Case 10

Test case 10 dilakukan pengujian dengan skenario 2 untuk pelanggan dengan id 18

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi kedua berdasarkan data riwayat pembelian Ibu HannyWijaya sebelumnya dan penilaian berdasarkan tipe pelanggan *catering*. Expert menilai bahwa semua produk memiliki kecenderungan yang tinggi untuk dibeli oleh customer dengan tipe *catering* seperti Ibu HannyWijaya

Tabel 6.28. Hasil uji set rekomendasi test case 10

cosine, with concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Bento Cutter - ORI A Japan Face	200.000	✓
2	Rice Mold Cute Cat with Nori Cutter	165.000	✓
3	Rice Mold – Segitiga	35.000	✓
4	Bento Cutter - Vegetable Disney Hello Kitty	95.000	✓
5	Rice Mold – GoPanda	150.000	✓
6	Die Cut Nori - Hello Kitty	30.000	✓
7	Catering Box Bento Hiro Mini LB110	2.500	✓

Dalam test case ini, tidak ditemukan 1 produkpun yang tidak sesuai sehingga persentase ketepatan rekomendasi dari test case 10 adalah 100% yang dapat dikategorikann sangat baik.

Tabel 6.29. Ringkasan hasil pengujian test case 10

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	0
Jumlah Rekomendasi yang benar	7
Persentase ketepatan rekomendasi	100%

6.1.4.4.11 Test Case 11

Test case 11 dilakukan pengujian dengan skenario 3 untuk pelanggan dengan id 18

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi pertama berdasarkan data riwayat pembelian Ibu HannyWijaya sebelumnya dan penilaian berdasarkan tipe pelanggan *catering*. Expert menilai bahwa produk no 2 dan 4 memiliki kemungkinan yang rendah untuk

dibeli bagi customer dengan tipe *catering* seperti Ibu HannyWijaya.

Tabel 6.30. Hasil uji set rekomendasi test case 11

euclidian, without concurrence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Rice Mold - Set Snoopy and Friends	55.000	✓
2	Sandwich mold - bear imotani	33.000	✗
3	Nori Punch - Face KAI A	160.000	✓
4	Cookie Cutter Thomas 4shape with stamp	60.000	✗
5	Bento Cutter - ORI A Japan Face	200.000	✓
6	Rice Mold - Set Animals Hello Kitty Small	35.000	✓
7	Rice Mold - Disney Ska Hello Kitty Medium	97.000	✓

Dalam test case ini, ditemukan 2 produk yang tidak sesuai dan 5 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 11 adalah 71% yang dapat dikategorikan baik.

Tabel 6.31. Ringkasan hasil pengujian test case 11

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	2
Jumlah Rekomendasi yang benar	5
Persentase ketepatan rekomendasi	71%

6.1.4.4.12 Test Case 12

Test case 12 dilakukan pengujian dengan skenario 4 untuk pelanggan dengan id 18

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi pertama berdasarkan data riwayat pembelian Ibu Hanny Wijaya sebelumnya dan penilaian berdasarkan tipe pelanggan *catering*. Expert menilai bahwa produk no 4 memiliki kemungkinan yang rendah untuk dibeli bagi customer dengan tipe *catering* seperti Ibu HannyWijaya.

Tabel 6.32. Hasil uji set rekomendasi test case 12

euclidian, with concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Rice Mold - Set Snoopy and Friends	55.000	✓
2	Bento Cutter - Alphabet ABC Torune	95.000	✓
3	Catering Box Bento Hiro LB-118 (BLACK)	3.000	✓
4	Sandwich mold - bear imotani	33.000	✗
5	Catering Box Bento Hiro Small LB115	3.000	✓
6	Die Cut Nori Mickey isi 50wajah	30.000	✓
7	Bento Book - Bento Karakter Unik (BKU)	60.000	✓

Dalam test case ini, ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 12 adalah 86% yang dapat dikategorikan baik.

Tabel 6.33. Ringkasan hasil pengujian test case 12

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.13 Test Case 13

Test case 13 dilakukan pengujian dengan skenario 1 untuk pelanggan dengan id 952

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi pertama berdasarkan data riwayat pembelian Ibu shintadwijayanti sebelumnya dan penilaian berdasarkan tipe pelanggan *personal-use*. Expert menilai bahwa seluruh produk memiliki kemungkinan yang tinggi untuk dibeli oleh customer dengan tipe *personal-use* seperti Ibu shintadwijayanti.

Tabel 6.34. Hasil uji set rekomendasi test case 13

cosine, without concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Rice Mold – Segitiga	35.000	✓
2	Pinset Ujung Bengkok	19.000	✓
3	Bento Cutter - Alphabet ABC Torune	95.000	✓
4	Sandwich cutter - Animals friend cutter kit	175.000	✓
5	Egg Mold - Box Car Fish	32.000	✓
6	Bento Book - Bento Karakter Unik (BKU)	60.000	✓
7	Sandwich Mold - Car Shape	33.000	✓

Dalam test case ini, seluruh produk dinilai benar sehingga persentase ketepatan rekomendasi dari test case 13 adalah 100% yang dapat dikategorikan sangat baik.

Tabel 6.35. Ringkasan hasil pengujian test case 13

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	0
Jumlah Rekomendasi yang benar	7
Persentase ketepatan rekomendasi	100%

6.1.4.4.14 Test Case 14

Test case 14 dilakukan pengujian dengan skenario 2 untuk pelanggan dengan id 952

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi pertama berdasarkan data riwayat pembelian Ibu shintadwijayanti sebelumnya dan penilaian berdasarkan tipe pelanggan *personal-use*. Expert menilai bahwa produk no 6 memiliki kemungkinan yang rendah untuk dibeli oleh customer dengan tipe *personal-use* seperti Ibu shintadwijayanti.

Tabel 6.36. Hasil uji set rekomendasi test case 14

cosine, with concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Sandwich cutter - Animals friend cutter kit	175.000	✓
2	Rice Mold – Segitiga	35.000	✓
3	Bento Cutter - Vegetable Disney Hello Kitty	95.000	✓
4	Nori Kemasan Ulang Takaoka	25.000	✓
5	Lunch Box - Set Piled Box Family Face	30.000	✓
6	Catering Box Bento Hiro Medium Sekat3 DB10B	3.800	✗
7	Cookie Cutter Winnie The Pooh	35.000	✓

Dalam test case ini, ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 14 adalah 86% yang dapat dikategorikan baik.

Tabel 6.37. Ringkasan hasil pengujian test case 14

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.15 Test Case 15

Test case 15 dilakukan pengujian dengan skenario 3 untuk pelanggan dengan id 952

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi ketiga berdasarkan data riwayat pembelian Ibu shintadwijayanti sebelumnya dan penilaian berdasarkan tipe pelanggan *personal-use*. Expert menilai bahwa produk no 2 memiliki kemungkinan yang rendah untuk dibeli oleh customer dengan tipe *personal-use* seperti Ibu shintadwijayanti.

Tabel 6.38. Hasil uji set rekomendasi test case 15

euclidian, without concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Nori Punch - Face KAI A	160.000	✓
2	Catering Box Bento Hiro Medium Sekat3 DB10B	3.800	✗
3	Sandwich mold - bear imotani	33.000	✓
4	Rice Mold - Set Animals Hello Kitty Small	35.000	✓
5	Cookie Cutter - Hello Kitty	35.000	✓
6	Cookie Cutter - Doraemon Dorami	35.000	✓
7	Cookie Cutter Winnie The Pooh	35.000	✓

Dalam test case ini, ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 15 adalah 86% yang dapat dikategorikan baik.

Tabel 6.39. Ringkasan hasil pengujian test case 15

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.1.4.4.16 Test Case 16

Test case 16 dilakukan pengujian dengan skenario 4 untuk pelanggan dengan id 952

Berikut ini merupakan penilaian expert terhadap ketepatan produk dari set rekomendasi keempat berdasarkan data riwayat pembelian Ibu shintadwijayanti sebelumnya dan penilaian berdasarkan tipe pelanggan *personal-use*. Expert menilai bahwa produk no 5 memiliki kemungkinan yang rendah untuk dibeli oleh customer dengan tipe *personal-use* seperti Ibu shintadwijayanti.

Tabel 6.40. Hasil uji set rekomendasi test case 16

euclidian, with concurence order			
No	Nama Produk Rekomendasi	Harga	Rekomendasi sesuai?
1	Pinset Ujung Bengkok	19.000	✓
2	Rice Mold - Set Animals Hello Kitty Small	35.000	✓
3	Nori Kemasan Ulang Takaoka	25.000	✓
4	Bento Cutter - Alphabet ABC Torune	95.000	✓
5	Rice Mold - Disney Ska Cars Tomica	97.000	✗
6	Egg Mold - Box Car Fish	32.000	✓
7	Sandwich mold - bear imotani	33.000	✓

Dalam test case ini, ditemukan 1 produk yang tidak sesuai dan 6 produk yang benar sehingga persentase ketepatan rekomendasi dari test case 16 adalah 86% yang dapat dikategorikan baik.

Tabel 6.41. Ringkasan hasil pengujian test case 16

Jumlah seluruh rekomendasi	7
Jumlah rekomendasi yang salah	1
Jumlah Rekomendasi yang benar	6
Persentase ketepatan rekomendasi	86%

6.2 Pembahasan

6.2.1 Analisis Hasil BlackBox testing

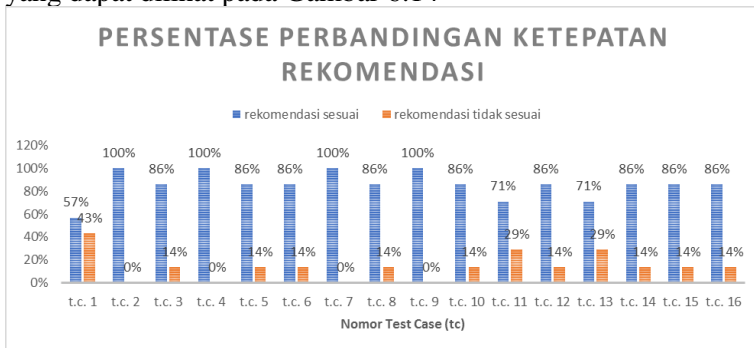
Pada bagian ini dilakukan pembahasan dan penyimpulan hasil dari serangkaian pengujian Black Box Testing yang telah dilakukan pada perangkat lunak di sisi client dan server. Black Box Testing dilakukan untuk menguji ketepatan aplikasi apakah hasil eksekusi dari aplikasi telah sesuai dengan apa yang telah di rancang sebelumnya. Pengujian juga dilakukan untuk menemukan kesalahan-kesalahan pada sistem dan struktur data, hal tersebut dilakukan untuk memastikan setiap fungsional aplikasi dapat berjalan dengan baik. Pengujian aplikasi

dilakukan berdasarkan hasil kebutuhan fungsional web service serta terpenuhinya kebutuhan fungsional dari sisi client. Hasil akhir pengujian memperlihatkan bahwasannya dari 15 Fitur yang dimiliki oleh aplikasi telah di uji dan dapat berjalan dengan baik tanpa ditemukan kendala atau kesalahan sistem.

6.2.2 Analisis Hasil Rekomendasi

6.2.2.1 Analisis Ketepatan Rekomendasi

Dari 16 hasil pengujian dapat diketahui masing-masing persentase kesesuaian set rekomendasi yang diberikan kepada pengguna. Penilaian dilakukan dengan menghitung persentase nilai ketepatan dan nilai kesalahan dari masing masing test case yang dapat dilihat pada Gambar 6.14



Gambar 6.14. Perbandingan ketepatan rekomendasi

Dari Gambar 6.14 dapat diketahui dari seluruh test case yang dilakukan, didapatkan tingkat ketepatan rekomendasi mencapai 85,8% sedangkan tingkat kesalahan rata-rata mencapai 14,1%. Hal ini menunjukkan bahwa penggunaan sistem rekomendasi dengan metode collaborative filtering secara keseluruhan dapat dinilai layak untuk diimplementasikan pada toko online forbento.

6.2.2.2 Analisis Perhitungan Rekomendasi

Saat pengujian, dilakukan percobaan pengaruh perhitungan kesamaan pelanggan dengan 2 metode yang berbeda yaitu menggunakan cosine similarity dan euclidian distance yang keduanya biasa diterapkan dalam algoritma collaborative filtering untuk. Pengujian ini dimaksudkan untuk mengetahui pola perhitungan yang baik untuk meningkatkan performa rekomendasi sehingga analisis kemiripan/ketidakmiripan ini dilakukan.

Dari setiap metode yang dilakukan didapatkan data sebagai berikut sebagai data perbandingan pada setiap perhitungan:

Tabel 6.42. Persentase hasil uji dengan cosine

Cosine similarity, without recurrence order			
test user 1	test user 2	test user 3	test user 4
57%	100%	86%	100%
Rata rata ketepatan			
0,8575			

Tabel 6.43. Persentase hasil uji dengan cosine & recurrence order

Cosine similarity , with recurrence order			
test user 1	test user 2	test user 3	test user 4
86%	86%	100%	86%
Rata rata ketepatan			
0,895			

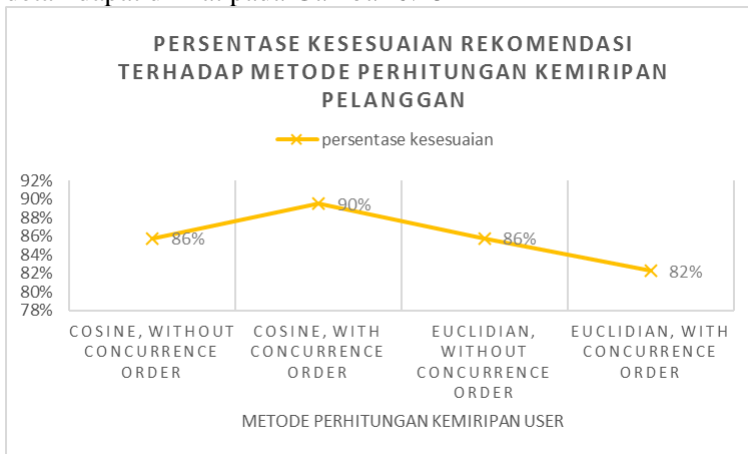
Tabel 6.44. Persentase hasil uji dengan euclidian

Euclidian distance, without recurrence order			
test user 1	test user 2	test user 3	test user 4
100%	86%	71%	86%
Rata rata ketepatan			
0,8575			

Tabel 6.45. Persentase hasil uji dengan euclidian & recurrence order

Euclidian distance, without recurrence order			
test user 1	test user 2	test user 3	test user 4
71%	86%	86%	86%
Rata rata ketepatan			
0,8225			

Dari data tersebut dapat diketahui bahwa rata rata perhitungan penggunaan cosine similarity lebih baik dengan rata-rata mencapai 87,6% daripada perhitungan menggunakan euclidian distance dengan nilai rata-rata mencapai 84% yang secara lebih detail dapat dilihat pada Gambar 6.15

**Gambar 6.15. Persentase Kesesuaian Rekomendasi terhadap metode perhitungan kemiripan pelanggan**

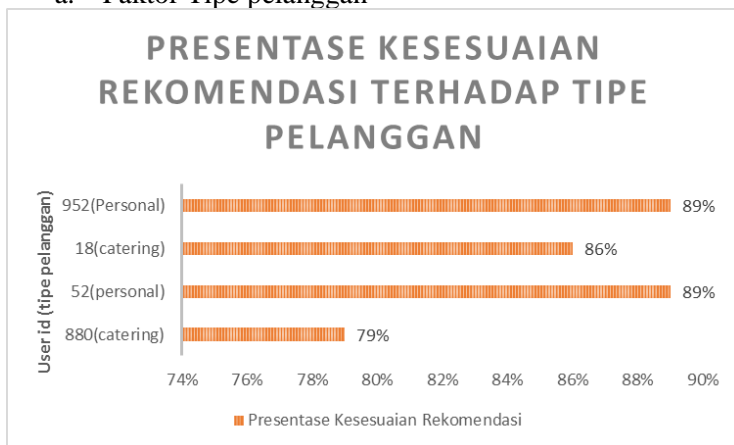
Namun, perbedaan ini tidak berbeda secara signifikan dan tidak berpengaruh banyak terhadap rekomendasi dalam studi kasus ini. Setelah dilakukan analisis pada hasil uji coba dan kedua metode perhitungan, kedua metode ini adalah fungsi untuk mencari perbedaan, diantara 2 buah vektor, namun yang membedakan kedua metode ini adalah cosine similarity mencari perbedaan sudut kedua vektor sedangkan euclidean distance

mencari jarak dari vektor dalam n-dimensi. Dengan algoritma yang disebutkan pada Gambar 6.15, cosine similarity cenderung tidak rentan terhadap perubahan panjang vektor selama rasio dari vektor tersebut tidak berubah sehingga tetap, metode cosine similarity lebih sesuai diterapkan dalam studi kasus rekomendasi collaborative filtering selama jumlah rasio yang digunakan dalam kasus ini (pelanggan, produk, dan penilaian/order) masih tetap, walaupun faktor lain dapat dimasukkan untuk meningkatkan performa rekomendasi.

6.2.2.3 Analisis Faktor Pengaruh Rekomendasi

Hasil rekomendasi yang dihasilkan dari pengujian, tidak terlepas dari beberapa faktor yang mampu mempengaruhi hasil rekomendasi secara mayor ataupun secara minor. Setiap e-commerce tentu memiliki pemahaman yang berbeda terhadap preferensi pelanggannya sesuai dengan tipe bisnis yang dijalankan. Dalam proses wawancara dengan pihak expert dari forbento, didapatkan beberapa faktor yang mendorong justifikasi pemilihan rekomendasi yang dijabarkan pada penjelasan di bawah ini:

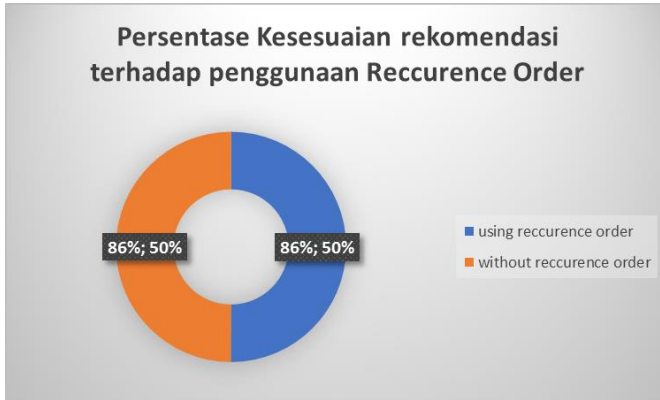
a. Faktor Tipe pelanggan



Gambar 6.16. pengaruh hasil rekomendasi terhadap tipe pelanggan

Dalam percobaan yang dilakukan dapat diketahui dari Gambar 6.16 bahwa rekomendasi yang dihasilkan lebih sesuai dengan tipe pelanggan personal dengan rata rata mencapai 89% daripada tipe pelanggan catering dengan rata-rata 82,5%.

b. Faktor Reccurence Order



Gambar 6.17. Persentase Kesesuaian rekomendasi terhadap penggunaan reccurence order

Recurrence order atau pembelian berulang digunakan untuk menggantikan nilai rating atau penilaian sebagai komponen penting sebuah rekomendasi. Namun dari penelitian ini dengan data yang didapatkan, diketahui bahwa pembeli cukup jarang melakukan pembelian berulang atau memiliki nilai reccurence order yang sama terhadap suatu produk. Hal ini menyebabkan sedikitnya rentang data yang dihasilkan dari similarity value atau derajat kemiripan antar user, dalam kasus ini hanya didapatkan rentang data 0.8164965809277261 – 1. Dengan demikian pemilihan rekomendasi terbaik bisa saja didapatkan dari banyak pelanggan yang juga memiliki derajat kemiripan yang sama. Untuk mendapatkan performa rekomendasi yang lebih baik bisa dilakukan dengan data yang lebih kaya untuk menilai preferensi pelanggan dari sudut pandang penjualan e commerce, seperti heatmap, clickstream, rating produk, dan lain-lain

Dari hasil percobaan dilakukan pengujian menggunakan 2 case yang berbeda yaitu penggunaan recurrence order (pembelian berulang) dan tanpa menggunakan recurrence order. Dari Gambar 6.17, dapat diketahui bahwa tidak ada perbedaan pada hasil rekomendasi dengan ada tidaknya faktor recurrence order yang disertakan dalam perhitungan.

Lebih jauh lagi, hasil wawancara yang didapatkan dari expert forbento juga menyebutkan bahwa penggunaan recurrence order tidak mempengaruhi kecenderungan pelanggannya untuk membeli barang yang memiliki tingkat recurrence order yang tinggi. Dari hal tersebut dapat disimpulkan bahwa hasil pengujian sesuai dengan hasil wawancara yang dilakukan.

6.2.3 Evaluasi

6.2.3.1 Evaluasi Pengujian

Evaluasi pengujian seharusnya dilakukan dengan metode decision support matriks dengan mempertimbangkan nilai precision dan recall. Untuk melakukan pengujian ketepatan rekomendasi, dilakukan pemisahan data dari setiap daftar transaksi yang pernah dilakukan pengguna. Dari transaksi tersebut, diambil sebagian data sebagai data testing dan sebagian data sebagai data training. Pengambilan data testing dipilih berdasarkan urutan pembelian oleh pelanggan pada periode yang terakhir. Seluruh data training dimodelkan dan dihitung kemiripannya. Hasil set rekomendasi untuk setiap pengguna dicocokkan dengan data testing dan dilakukan observasi terhadap hasil rekomendasi yang sama dengan data testing yang merupakan data transaksi pelanggan yang sebenarnya. Dari set rekomendasi yang dihasilkan dapat dinilai akurasi rekomendasi berdasarkan data pembelian pelanggan selanjutnya pada data testing.

6.2.3.2 Evaluasi Penilaian Kemiripan

Dalam penelitian ini, penggunaan rumus cosine similarity dan euclidian distance untuk mengetahui kemiripan pengguna tidak mempertimbangkan perbedaan pembelian antara kedua pelanggan dan hanya mempertimbangkan kesamaan dari kedua pelanggan. Oleh karena itu, jika terdapat perbedaan antara pelanggan yang pernah membeli suatu barang atau tidak pernah membeli suatu barang, maka nilai yang diberikan dalam penelitian ini adalah nilai null, bukan nilai 0.

Terdapat beberapa kekurangan dalam penerapan hal ini seperti nilai similarity pada graf kurang baik dalam merepresentasikan kedekatan pelanggan karena nilai kemiripan yang tinggi tidak selalu menggambarkan preferensi yang sama sehingga hasil yang didapatkan cenderung tidak bervariasi.

Namun, jika digunakan alternatif lain untuk mengatasinya yaitu dengan memberikan nilai 0 pada setiap relationship antara pelanggan dan barang secara default saat pelanggan tersebut belum pernah melakukan pembelian, akan berdampak pada penurunan efisiensi graf karena semakin banyaknya kapasitas memori untuk permodelan graf yang digunakan.

Dari hal tersebut, penggunaan recurrence order dinilai kurang tepat dalam kasus ini mengingat recurrence order tidak merepresentasikan kesukaan/ketidaksukaan pelanggan terhadap setiap barang. Penggunaan rating terhadap barang lebih disarankan untuk menjadi penilaian kemiripan pelanggan.

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Setelah melakukan implementasi dan pengujian pada penelitian “Rancang Bangun web service rekomendasi dengan database neo4j berbasis graph”, dapat disimpulkan beberapa hal sebagai berikut:

1. Web Service rekomendasi dibangun dalam lingkungan teknologi node js dengan menggunakan database graph neo4j melalui koneksi bolt. Perancangan meliputi pengolahan data yang menghasilkan 963 data pelanggan, 277 produk dan 2541 transaksi yang selanjutnya diimplementasikan dalam model graf. Sistem client dibangun dengan mengimplementasikan 3 rancangan use case yang dibuat sendiri, dan 7 implementasi use case yang telah tersedia dalam framework wordpress. Sistem server dibangun dengan mengimplementasikan 15 rancangan use case yang dibangun dengan bahasa javascript dan framework express node
2. Permodelan data graph pada web service ini terdiri dari 4 buah node yaitu, Member, Shop, Products, dan Customer dengan 3 jenis relationship yaitu own, buy, dan similarity. Dengan pembentukan hubungan similarity antar user, hubungan pembelian antara pelanggan dan product serta hubungan kepemilikan antara member dan shop.
3. Dengan permodelan yang digunakan dalam penelitian ini, jumlah waktu rata-rata yang dibutuhkan untuk membuat seluruh similarity dari 963 pelanggan dibutuhkan waktu 23715 ms, sedangkan untuk mendapatkan rekomendasi produk melalui database graf neo4j adalah 7 ms.

4. Kode untuk perhitungan kemiripan pelanggan yang digunakan dalam penelitian ini sudah valid dan sesuai dengan rumus cosine similarity maupun euclidian distance
5. Penggunaan recurrence order sebagai pengganti preferensi pelanggan tidak cukup berpengaruh dalam menghasilkan nilai derajat kemiripan pelanggan.
6. Spesifikasi lingkungan implementasi yang digunakan pada penelitian ini kurang baik untuk menangani visualisasi graf neo4j dengan alokasi penyimpanan neo4j yang tersisa sebesar 109.87 MiB, yakni terdiri dari jumlah node yang teralokasi adalah 10002, jumlah property 22250, jumlah relationship 56421, dan 5 tipe relationship.
7. Pengujian hasil rekomendasi dilakukan dengan jumlah 16 skenario dan 4 pengguna, dan menghasilkan tingkat kesesuaian rekomendasi mencapai 85,8% dan tingkat kesalahan mencapai 14,1% dengan pengujian menggunakan justifikasi expert
8. Rata rata perhitungan penggunaan cosine similarity lebih baik dengan rata-rata mencapai 87,6% daripada perhitungan menggunakan euclidian distance dengan nilai rata-rata mencapai 84% dalam mencari tingkat kemiripan antar user dalam algoritma collaborative filtering dengan pengujian menggunakan justifikasi expert

7.2 Saran

Saran penulis yang dapat dilakukan untuk penelitian selanjutnya ialah:

1. Pengujian sebaiknya dilakukan dengan metode decision support matriks dengan mempertimbangkan nilai recall dan precision dan pemisahan data permodelan dan data testing berdasarkan periode waktu/ urutan pembelian.

2. Penambahan data untuk merepresentasikan preferensi pelanggan terhadap suatu produk dalam bentuk skala numerik (rating produk).
3. Penggunaan lingkungan implementasi dengan spesifikasi yang lebih tinggi untuk memvisualisasikan graph pada penelitian ini.
4. Pengembangan sistem untuk menangani proses pengambilan data dan pembuatan API dari jenis client yang berbeda.
5. Pembuatan layer data khusus pada server untuk menangani berbagai perbedaan jenis data json yang akan diterima oleh client.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] APJII, “Data dan Statistik KOMINFO | Data Browser,” 2016. [Online]. Available: <http://statistik.kominfo.go.id/site/data?idtree=424&iddoc=1517>. [Accessed: 13-Feb-2017].
- [2] “Membuka dan Mengembangkan Potensi E-Commerce di Indonesia – Presiden Republik Indonesia,” 2016. [Online]. Available: <http://presidenri.go.id/topik-aktual/membuka-dan-mengembangkan-potensi-e-commerce-di-indonesia.html>. [Accessed: 13-Feb-2017].
- [3] Datanyze, “E-commerce Platforms Market Share Report | Competitor Analysis in Indonesia | Datanyze,” 2017.
- [4] H. Chen, M. Gan, and M. Song, “A Graph Model for Recommender Systems,” *Proc. 2nd Int. Conf. Comput. Sci. Electron. Eng. (ICCSEE 2013)*, vol. 55, no. Iccsee, pp. 878–881, 2013.
- [5] D. E. Wibowo, “Sistem Rekomendasi Jual Beli Barang Dengan Memanfaatkan Metode Collaborative Filtering dan Basis Data Graf . Studi Kasus : Bukalapak . com,” 2013.
- [6] Neo4j, “neo4j: World’s Leading Graph Database,” *Neo4J.Org*, 2012. [Online]. Available: <http://neo4j.org/>.
- [7] A. Sharma, “Enhancing the Accuracy of Recommender System Using Graph Databases,” no. July, 2015.
- [8] J. Manajemen and I. D. Teknoligi, “Expert – Jurnal Manajemen sistem Informasi Dan Teknoligi.”
- [9] P. F. Tanaem, “RESTFul Web Service Untuk Sistem Pencatatan,” pp. 2443–2229, 2016.
- [10] afriansyah dwi putra, “Pengertian SaaS, PaaS dan IaaS | Zwei Messerschmitt.” [Online]. Available: <https://zweimesserschmitt.wordpress.com/2012/02/24/pengertian-saas-paas-dan-iaas/>. [Accessed: 28-Feb-2017].
- [11] Abhishek Sur, “Building a CRUD in RESTful Services of WCF: October 2010,” 2010. [Online]. Available: http://www.abhisheksur.com/2010_10_01_archive.htm

1. [Accessed: 28-Feb-2017].
- [12] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in *Recommender Systems Handbook*, vol. 54, no. OCTOBER, 2011, pp. 1–35.
- [13] S. C. Kabore, "Design and implementation of a recommender system as a module for Liferay portal," vol. 1, no. September, pp. 153–162, 2012.
- [14] G. Singh and R. Singh Boparai, "A survey on recommendation system," *IOSR J. Comput. Eng. Ver. V*, vol. 17, no. 6, pp. 2278–661, 2015.
- [15] Y. Song, S. Dixon, and M. Pearce, "A Survey of Music Recommendation Systems and Future Perspectives," in *9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)*, 2012.
- [16] C. Yu, Y. Luo, and K. Liu, "Multi-attribute Collaborative Filtering Recommendation Descriptions of Basic Model User rating matrix," vol. 10, no. 5, pp. 746–759, 2015.
- [17] Badrul M. Sarwar, "Overview of the Collaborative Filtering Process," 2001. [Online]. Available: <http://www10.org/cdrom/papers/519/node6.html>. [Accessed: 28-Feb-2017].
- [18] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proc. 10th ...*, vol. 1, pp. 285–295, 2001.
- [19] C. A. Levinas, "Master of Science Thesis An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals," no. September, 2014.
- [20] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," *Proc. 10th ...*, vol. 1, pp. 285–295, 2001.
- [21] M. Gjoka and F. Soldo, "Exploring collaborative filters : Neighborhood-based approach," *Methods*, pp. 1–7, 2005.
- [22] Sendhy Rachmat Wurdianarto, "PERBANDINGAN

EUCLIDEAN DISTANCE DENGAN CANBERRA DISTANCE PADA FACE RECOGNITION,” *Techno.COM*, vol. 13, 2014.

- [23] B. J. Mirza, B. J. Keller, and N. Ramakrishnan, “Studying recommendation algorithms by graph analysis,” *J. Intell. Inf. Syst.*, vol. 20, no. 2, pp. 131–160, 2003.
- [24] Neo4j, “Powering Recommendations with a Graph Database.”
- [25] G. Maps, “About the Tutorial Execute Node . js Online.”
- [26] E. P. Cundomanik, L. P. Dewi, and J. Andjarwirawan, “Aplikasi Survei Berbasis Android,” *Fax*.
- [27] Raman Ramsin, “Software Development Methodologies.”
- [28] Z. Huang, W. Chung, and H. Chen, “A graph model for e-commerce recommender systems,” *J. Am. Soc. Inf. Sci. Technol.*, vol. 55, no. 3, pp. 259–274, 2004.

BIODATA PENULIS



Penulis lahir pada tanggal 4 Maret 1995 di Bangkalan. Merupakan anak pertama dari 4 bersaudara. Penulis telah menempuh beberapa pendidikan formal yaitu; SDN Kaliasin VIII Surabaya, SMP Negeri 4 Surabaya, dan SMA Negeri 9 Surabaya.

Pada tahun 2013 pasca kelulusan SMA, penulis melanjutkan pendidikan dengan jalur SNMPTN (Undangan) di Jurusan Sistem Informasi FTIf – Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 5213100090. Selama menjadi mahasiswa, penulis mengikuti berbagai kegiatan kemahasiswaan seperti beberapa kepanitiaan seperti Information System Expo serta pernah menjabat sebagai Kepala Departemen Inovasi Karya Himpunan Mahasiswa Sistem Informasi pada kepengurusan Muda Berkarya. Selain itu, kegiatan seperti Latihan Ketrampilan Manajemen Mahasiswa pun pernah diikuti. Di bidang akademik, penulis aktif menjadi asisten dosen dan asisten praktikum pada beberapa mata kuliah seperti Matematika Diskrit dan Desain Basis Data.

Pada tahun keempat, karena penulis memiliki ketertarikan di bidang pengolahan data, maka penulis mengambil bidang minat Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui *email* di marinavisafitri@gmail.com.

A. LAMPIRAN A

A.1. Lembar Persetujuan Narasumber

Pendahuluan

Testing ini dilakukan untuk melakukan penilaian terhadap akurasi rekomendasi dari aplikasi yang akan dibuat oleh peneliti.

Aplikasi yang akan dibuat diberi nama Recommence. Recommence merupakan web service yang membantu e-commerce memproses rekomendasi berdasarkan kemiripan pelanggannya dari data data yang dimilikinya.

Proses pengujian ini akan memakan waktu sekitar 20 menit dan media yang digunakan adalah aplikasi *Minimum Viable Product*. Harapannya hasil masukan dari *testing* akan dibuat untuk mengembangkan aplikasi kedepannya, seluruh fitur dan fungsi dari aplikasi dibuat berdasarkan literatur dan penelitian sebelumnya.

Terdapat 2 Bagian pertanyaan yang harus dijawab. Pada bagian pertama menggunakan tipe soal dengan perbandingan dengan likert skala 1 – 5 yang bertujuan mengukur keyakinan expert terhadap aplikasi. Pada bagian kedua menggunakan tipe soal bebas, untuk mengambil evaluasi dan saran untuk pengembangan aplikasi.


Terima kasih atas partisipasi anda.

FORM PERSETUJUAN

Saya mengerti bahwa partisipasi dalam studi kegunaan ini bersifat sukarela.

Silakan isi form berikut untuk menunjukkan bahwa Anda telah membaca dan Anda memahami informasi pada formulir ini dan bahwa setiap pertanyaan yang mungkin Anda miliki tentang sesi telah dijawab.

Tanggal : 2 Juli 2019 .
Nama Lengkap : Rahayu Fatmawati
Jenis Kelamin : Perempuan
Usia : 32 thn .
Role : Owner .

Tanda tangan : 

Gambar A.1. Lembar Persetujuan Narasumber

A.2. Pengujian user 1

User 1:

Ibu Fera merupakan salah satu pelanggan Forbento yang cukup loyal, dari data pembelian Ibu Fera telah melakukan 5 kali transaksi

Berikut ini merupakan barang-barang yang dibeli sebelumnya:

1. Rice Mold - Disney Ska Cars Tomica - Rp. 97.000
2. Rice Mold - Disney Ska Winnie the Pooh - Rp. 97.000
3. Catering Box Bento Hiro Medium Sekat DB10B - Rp. 3500
4. Catering Box Hiro For Sushi ST-2 - Rp. 3300
5. Nori Kemasan Ulang Takaoka - Rp. 25.000

Hasil Rekomendasi:

Set 1: cosine, without order value

1. Die Cut Nori Face KAI isi 50wajah - 30.000
- ☒ Sandwich mold - bear imotani - 33.000
- ☒ Sandwich mold - Persegi Tutup - 33.000
- ☒ Sandwich mold - Persegi Tutup - 33.000
5. Nori Punch - Face KAI A - 160.000
6. Catering Box Bento Hiro Medium Tanpa Sekat DB10A - 3500
7. Sandwich Cutter - Hello Kitty Sanrio - 40.000

Set 2: Cosine, with order value

1. Bento Cutter - ORI A Japan Face - 200.000
- ☒ Sandwich Cutter - Animals Toy Tsurumi - 135000
3. Rice Mold Cute Cat with Nori Cutter - 165000
4. Rice Mold - Segitiga - 35000
5. Rice Mold - GoPanda - 150000
6. Die Cut Nori - Hello Kitty - 30000
7. Catering Box Bento Hiro Small LB115 - 3000

Set 3: euclidian asc

1. Rice Mold - Disney Ska Lilo Stitch - 97k
2. Cookie Cutter - Lilo and Stitch - 35k
3. Rice Mold - Set Animals Hello Kitty - 35k
4. Cookie Cutter - Doraemon Dorami - 35k
5. Rice Mold - Disney Ska Mickey Mouse - 97k
6. Cookie Cutter Marie Cat - 35k
7. Rice Mold - Set Snoopy and Friends - 55k

Set 4: euclidian desc

1. Die Cut Nori Face KAI isi 50wajah - 30k
- ☒ Sandwich mold - bear imotani - 33k
- ☒ Sandwich mold - Persegi Tutup - 33k
4. Rice Mold - Catatan Nasi Curry Hello Kitty - 160k
5. Rice Mold - Catatan Nasi Curry Rilakuma - 160k
6. Rice Mold - Catatan Nasi Curry Snoopy - 160k
7. Die Cut Nori Winnie the Pooh by Face KAI - 30k

Reasoning: Catering

Gambar A.2. Lembar Pengujian user 1

A.3. Pengujian user 2

User 2:

History Pembelian Ibu Winda(52):

1. Rice Mold - Set Animals Hello Kitty Small – 35.000
2. Sandwich mold - bear imotani – 33.000
3. Egg Mold - Box Car Fish – 32.000
4. Rice Mold - Set Snoopy and Friends – 55.000
5. Nori Kemasan Ulang Takaoka – 25.000

Set 1: cosine, without order value

1. Pinset Ujung Bengkok – 19.000
2. Bento Book - Bento Karakter Unik (BKU) – 60.000
3. Sandwich Mold - Car Shape – 33.000
4. Cookie Cutter - Mickey Minnie Mouse – 35.000
5. Egg Mold - Box Rabbit Bear – 32.000
6. Bento Cutter - ORI A Japan Face – 200.000
7. Rice Mold - Disney Ska Cars Tomica – 97.000

Set 2: Cosine, with order value

1. Bento Cutter - ORI A Japan Face – 200.000
2. Sandwich Cutter - Animals Toy Torune – 135000
3. Bento Cutter - Mold Set Animals Elephant – 80000
4. Sandwich cutter - Animals friend cutter kit – 175.000
5. Rice Mold Cute Cat with Nori Cutter – GoPanda – 165.000
6. Rice Mold - Segitiga – 35000
7. Bento Cutter - Vegetable Disney Hello Kitty - 95000

Set 3: euc, asc

1. Sandwich Mold - Car Shape – 33k
2. Catering Box Bento Hiro Medium Sekat3 DB10B-3800
3. Rice Mold - Disney Ska Lilo Stitch -97k
4. Cookie Cutter - Lilo and Stitch – 35k
5. Cookie Cutter - Doraemon Dorami – 35k
6. Rice Mold - Disney Ska Winnie the Pooh – 97k
7. Rice Mold - Disney Ska Mickey Mouse -97k

Set 4: euc, desc

1. Bento Book - iBento Indonesian Bento Book kayleen mami – 80k
2. Nori Punch - Lotus Flower – 25k
3. Rice Mold - Disney Ska Winnie the Pooh – 97k
4. Sandwich cutter - Animals friend cutter kit – 175k
5. Cookie Cutter - Hello Kitty 4shape with stamp – 60k
6. Catering Box Bento Hiro Medium Sekat3 DB10B – 3800
7. Nori Punch - Face KAI A – 160k

Reasoning: Personal

Gambar A.3. Lembar Pengujian User 3

A.4. Pengujian user 3

User 3:

History Pembelian Bu HannyWijaya (18)

1. Pinset Ujung Bengkok - 19.000
2. Catering Box Bento Hiro Medium Sekali (DR108 - 3800 (15 item)
3. Bento Cutter - Arrange Set Torune - 120000
4. Nori Kemasan Ulang Takaka - 25.000

Set 1: cosine, without order value

1. Sandwich Mold - Car Shape - 33000
2. Die Cut Nori Mickey isi 50wajah - 30000
3. Rice Mold - Disney Ska Mickey Mouse - 97000
4. Die Cut Nori Face KAJ isi 50wajah - 30000
5. Die Cut Nori - Hello Kitty - 30000
6. Rice Mold - Tor Four Animals - 80000
- ~~7. Lunch Pick Eyess - 60000~~

Set 1: cosine, without order value

1. Bento Cutter - ORI B Japan Face - 200000
2. Rice Mold Cute Cat with Nori Cutter - 165000
3. Rice Mold - Segitiga - 35000
4. Bento Cutter - Vegetable Disney Hello Kitty - 95000
5. Rice Mold - GoPanda - 150000
6. Die Cut Nori - Hello Kitty - 30000
7. Catering Box Bento Hiro Mini LB110 - 2500

Set 3: eur, asc

1. Rice Mold - Set Snoopy and Friends - 55k
- ~~2. Sandwich mold - bear imotani - 33k~~
3. Nori Punch - Face KAJ A - 160k
- ~~4. Cookie Cutter Thomas 4shape with stamp - 60k~~
5. Bento Cutter - ORI A Japan Face - 200k
6. Rice Mold - Set Animals Hello Kitty Small - 35k
7. Rice Mold - Disney Ska Hello Kitty Medium - 97k

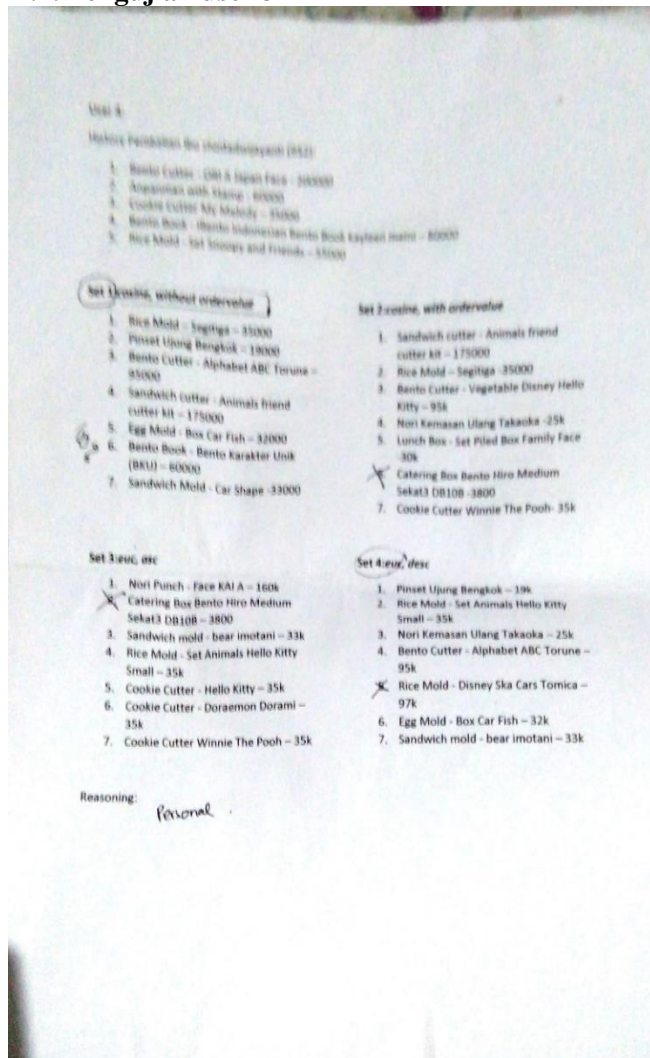
Set 4: eur, desc

1. Rice Mold - Set Snoopy and Friends - 55k
2. Bento Cutter - Alphabet ABC Torune - 95k
3. Catering Box Bento Hiro LB-118 (BLACK) - 3k
- ~~4. Sandwich mold - bear imotani~~
5. Catering Box Bento Hiro Small LB115 - 3k
6. Die Cut Nori Mickey isi 50wajah - 30k
7. Bento Book - Bento Karakter Unik (BIKU) - 60k

Reasoning: *Cat*

Gambar A.4. Lembar Pengujian User 4

A.4. Pengujian user 3



Gambar A.5. Lembar pengujian user 4